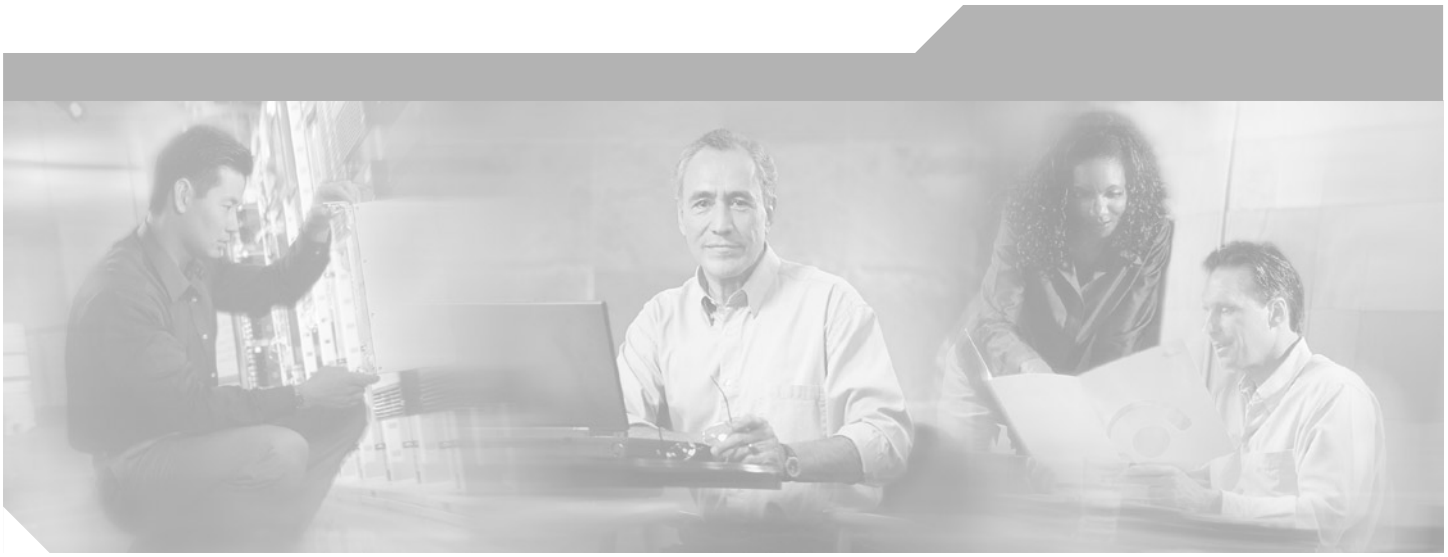


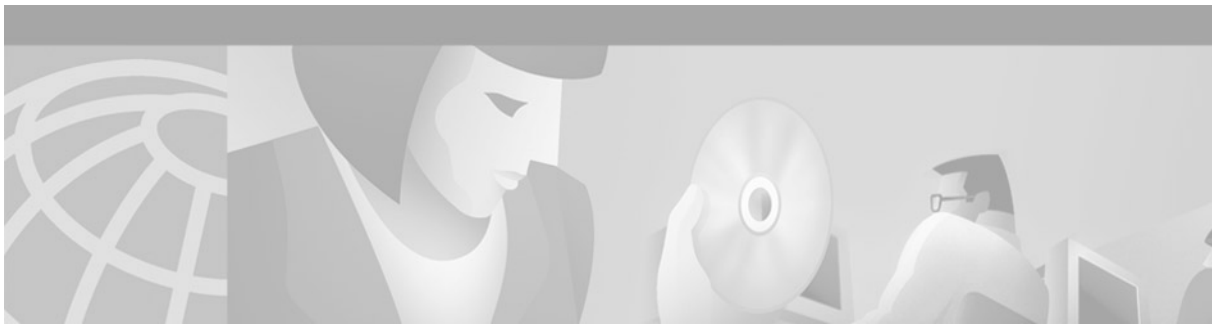
Cisco IOS
Mobile Wireless Configuration Guide

GGSN 4.0

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number:

CONTENTS



About Cisco IOS Software Documentation

This chapter discusses the objectives, audience, organization, and conventions of Cisco IOS software documentation. It also provides sources for obtaining documentation from Cisco Systems.

Documentation Objectives

Cisco IOS software documentation describes the tasks and commands necessary to configure and maintain Cisco networking devices.

Audience

The Cisco IOS software documentation set is intended primarily for users who configure and maintain Cisco networking devices (such as routers and switches) but who may not be familiar with the tasks, the relationship between tasks, or the Cisco IOS software commands necessary to perform particular tasks. The Cisco IOS software documentation set is also intended for those users experienced with Cisco IOS software who need to know about new features, new configuration options, and new software characteristics in the current Cisco IOS software release.

Documentation Organization

The Cisco IOS software documentation set consists of documentation modules and master indexes. In addition to the main documentation set, there are supporting documents and resources.

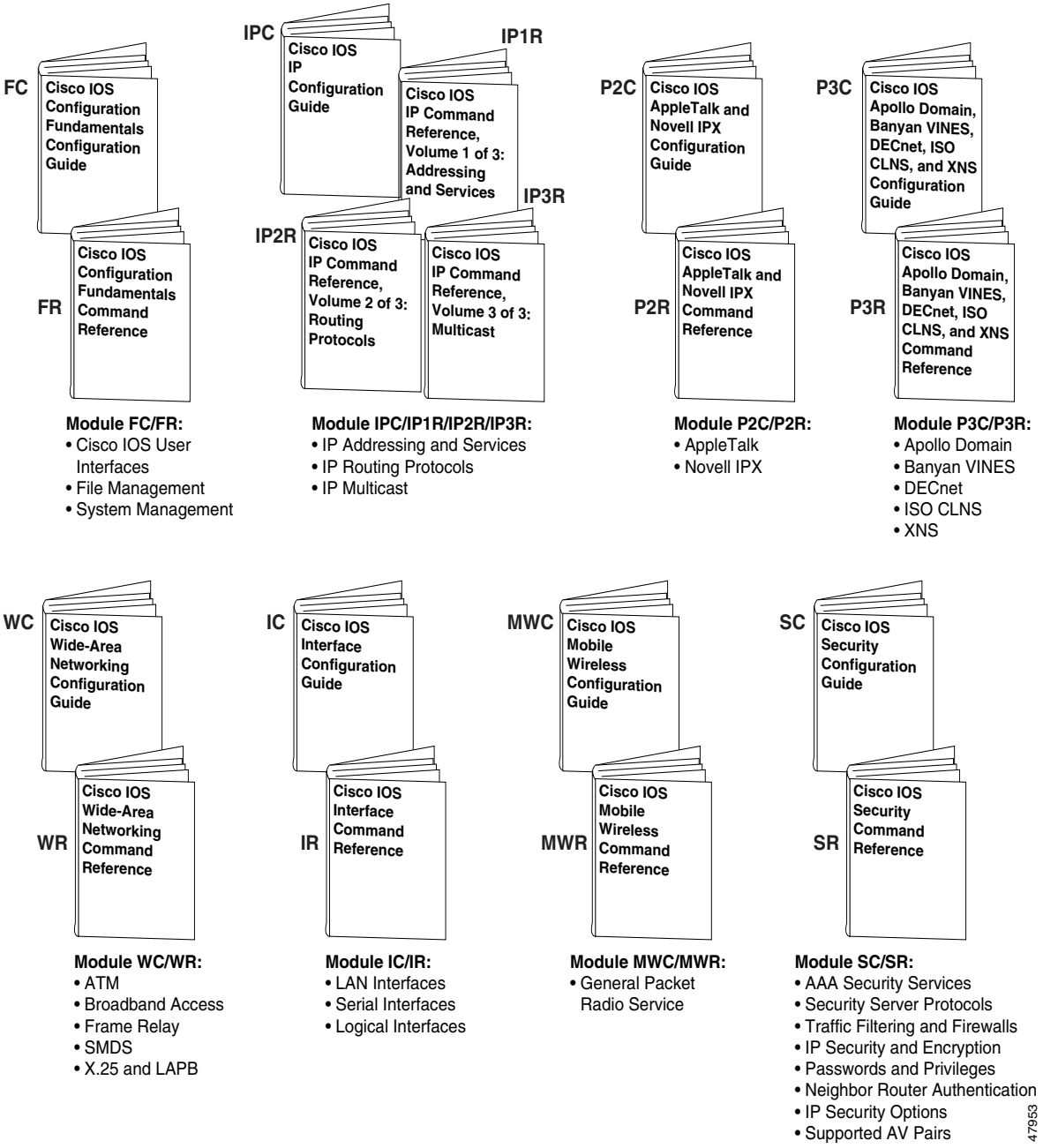
Documentation Modules

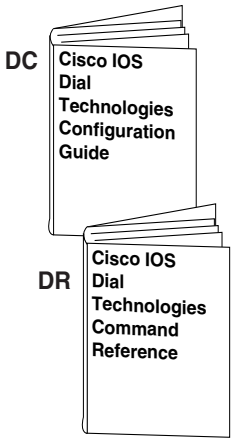
The Cisco IOS documentation modules consist of configuration guides and corresponding command reference publications. Chapters in a configuration guide describe protocols, configuration tasks, and Cisco IOS software functionality and contain comprehensive configuration examples. Chapters in a command reference publication provide complete Cisco IOS command syntax information. Use each configuration guide in conjunction with its corresponding command reference publication.

Figure 1 shows the Cisco IOS software documentation modules.

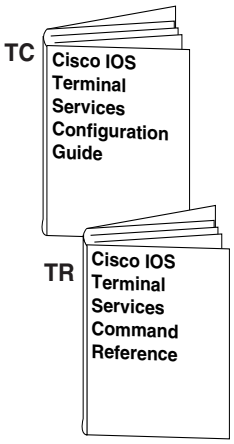
Note The abbreviations (for example, FC and FR) next to the book icons are page designators, which are defined in a key in the index of each document to help you with navigation. The bullets under each module list the major technology areas discussed in the corresponding books.

Figure 1 Cisco IOS Software Documentation Modules

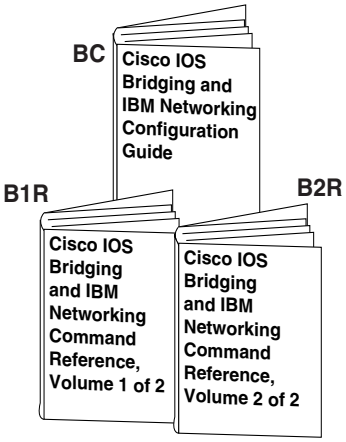




Module DC/DR:
- Preparing for Dial Access
- Modem and Dial Shelf Configuration and Management
- ISDN Configuration
- Signalling Configuration
- Dial-on-Demand Routing Configuration
- Dial-Backup Configuration
- Dial-Related Addressing Services
- Virtual Templates, Profiles, and Networks
- PPP Configuration
- Callback and Bandwidth Allocation Configuration
- Dial Access Specialized Features
- Dial Access Scenarios

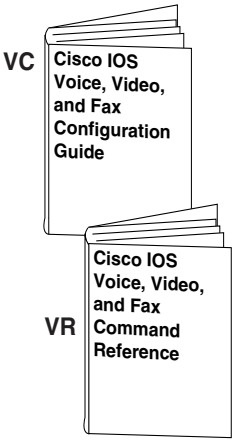


Module TC/TR:
- ARA
- LAT
- NASI
- Telnet
- TN3270
- XRemote
- X.28 PAD
- Protocol Translation

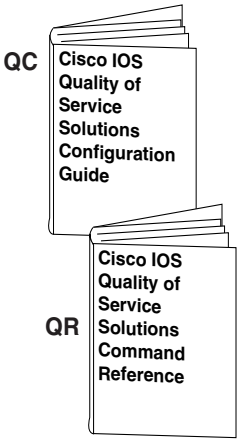


Module BC/B1R:
- Transparent Bridging
- SRB
- Token Ring Inter-Switch Link
- Token Ring Route Switch Module
- RSRB
- DLSw+
- Serial Tunnel and Block Serial Tunnel
- LLC2 and SDLC
- IBM Network Media Translation
- SNA Frame Relay Access
- NCIA Client/Server
- Airline Product Set

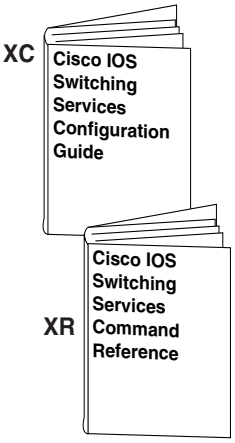
Module BC/B2R:
- DSPU and SNA Service Point
- SNA Switching Services
- Cisco Transaction Connection
- Cisco Mainframe Channel Connection
- CLAW and TCP/IP Offload
- CSNA, CMPC, and CMPC+
- TN3270 Server



Module VC/VR:
- Voice over IP
- Call Control Signalling
- Voice over Frame Relay
- Voice over ATM
- Telephony Applications
- Trunk Management
- Fax, Video, and Modem Support



Module QC/QR:
- Packet Classification
- Congestion Management
- Congestion Avoidance
- Policing and Shaping
- Signalling
- Link Efficiency Mechanisms



Module XC/XR:
- Cisco IOS Switching Paths
- NetFlow Switching
- Multiprotocol Label Switching
- Multilayer Switching
- Multicast Distributed Switching
- Virtual LANs
- LAN Emulation

47954

Master Indexes

Two master indexes provide indexing information for the Cisco IOS software documentation set: an index for the configuration guides and an index for the command references. Individual books also contain a book-specific index.

The master indexes provide a quick way for you to find a command when you know the command name but not which module contains the command. When you use the online master indexes, you can click the page number for an index entry and go to that page in the online document.

Supporting Documents and Resources

The following documents and resources support the Cisco IOS software documentation set:

- *Cisco IOS Command Summary* (two volumes)—This publication explains the function and syntax of the Cisco IOS software commands. For more information about defaults and usage guidelines, refer to the Cisco IOS command reference publications.
- *Cisco IOS System Error Messages*—This publication lists and describes Cisco IOS system error messages. Not all system error messages indicate problems with your system. Some are purely informational, and others may help diagnose problems with communications lines, internal hardware, or the system software.
- *Cisco IOS Debug Command Reference*—This publication contains an alphabetical listing of the **debug** commands and their descriptions. Documentation for each command includes a brief description of its use, command syntax, usage guidelines, and sample output.
- *Dictionary of Internetworking Terms and Acronyms*—This Cisco publication compiles and defines the terms and acronyms used in the internetworking industry.
- New feature documentation—The Cisco IOS software documentation set documents the mainline release of Cisco IOS software (for example, Cisco IOS Release 12.2). New software features are introduced in early deployment releases (for example, the Cisco IOS "T" release train for 12.2, 12.2(x)T). Documentation for these new features can be found in standalone documents called "feature modules." Feature module documentation describes new Cisco IOS software and hardware networking functionality and is available on Cisco.com and the Documentation CD-ROM.
- Release notes—This documentation describes system requirements, provides information about new and changed features, and includes other useful information about specific software releases. See the section "Using Software Release Notes" in the chapter "Using Cisco IOS Software" for more information.
- Caveats documentation—This documentation provides information about Cisco IOS software defects in specific software releases.
- RFCs—RFCs are standards documents maintained by the Internet Engineering Task Force (IETF). Cisco IOS software documentation references supported RFCs when applicable. The full text of referenced RFCs may be obtained on the World Wide Web at http://www.rfc-editor.org/.
- MIBs—MIBs are used for network monitoring. For lists of supported MIBs by platform and release, and to download MIB files, see the Cisco MIB website on Cisco.com at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

Document Conventions

Within Cisco IOS software documentation, the term *router* is generally used to refer to a variety of Cisco products (for example, routers, access servers, and switches). Routers, access servers, and other networking devices that support Cisco IOS software are shown interchangeably within examples. These products are used only for illustrative purposes; that is, an example that shows one product does not necessarily indicate that other products are not supported.

The Cisco IOS documentation set uses the following conventions:

Convention	Description
^ or Ctrl	The ^ and Ctrl symbols represent the Control key. For example, the key combination ^D or Ctrl-D means hold down the Control key while you press the D key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to public, do not use quotation marks around the string or the string will include the quotation marks.

Command syntax descriptions use the following conventions:

Convention	Description
boldface	Boldface text indicates commands and keywords that you enter literally as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example:

Convention	Description
[x {y | z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
`screen`	Examples of information displayed on the screen are set in Courier font.
`boldface screen`	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords.

Convention	Description
!	An exclamation point at the beginning of a line indicates a comment line. (Exclamation points are also displayed by the Cisco IOS software for certain processes.)
[ ]	Square brackets enclose default responses to system prompts.

The following conventions are used to attract the attention of the reader:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials not contained in this manual.

Timesaver Means the *described action saves time*. You can save time by performing the action described in the paragraph.

Obtaining Documentation

The following sections provide sources for obtaining documentation from Cisco Systems.

World Wide Web

The most current Cisco documentation is available on the World Wide Web at the following website:

http://www.cisco.com

Translated documentation is available at the following website:

http://www.cisco.com/public/countries_languages.html

Documentation CD-ROM

Cisco documentation and additional literature are available in a CD-ROM package, which ships with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or through an annual subscription.

Ordering Documentation

Cisco documentation can be ordered in the following ways:

- Registered Cisco Direct Customers can order Cisco product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/cgi-bin/order/order_root.pl

- Registered Cisco.com users can order the Documentation CD-ROM through the online Subscription Store:

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

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco corporate headquarters (California, USA) at 408 526-7208 or, in North America, by calling 800 553-NETS(6387).

Documentation Feedback

If you are reading Cisco product documentation on the World Wide Web, you can submit technical comments electronically. Click **Feedback** in the toolbar and select **Documentation**. After you complete the form, click **Submit** to send it to Cisco.

You can e-mail your comments to bug-doc@cisco.com.

To submit your comments by mail, use the response card behind the front cover of your document, or write to the following address:

Cisco Systems, Inc.
Document Resource Connection
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com as a starting point for all technical assistance. Customers and partners can obtain documentation, troubleshooting tips, and sample configurations from online tools. For Cisco.com registered users, additional troubleshooting tools are available from the TAC website.

Cisco.com

Cisco.com is the foundation of a suite of interactive, networked services that provides immediate, open access to Cisco information and resources at anytime, from anywhere in the world. This highly integrated Internet application is a powerful, easy-to-use tool for doing business with Cisco.

Cisco.com provides a broad range of features and services to help customers and partners streamline business processes and improve productivity. Through Cisco.com, you can find information about Cisco and our networking solutions, services, and programs. In addition, you can resolve technical issues with online technical support, download and test software packages, and order Cisco learning materials and merchandise. Valuable online skill assessment, training, and certification programs are also available.

Customers and partners can self-register on Cisco.com to obtain additional personalized information and services. Registered users can order products, check on the status of an order, access technical support, and view benefits specific to their relationships with Cisco.

To access Cisco.com, go to the following website:

http://www.cisco.com

Technical Assistance Center

The Cisco TAC website is available to all customers who need technical assistance with a Cisco product or technology that is under warranty or covered by a maintenance contract.

Contacting TAC by Using the Cisco TAC Website

If you have a priority level 3 (P3) or priority level 4 (P4) problem, contact TAC by going to the TAC website:

http://www.cisco.com/tac

P3 and P4 level problems are defined as follows:

- P3—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- P4—You need information or assistance on Cisco product capabilities, product installation, or basic product configuration.

In each of the above cases, use the Cisco TAC website to quickly find answers to your questions.

To register for Cisco.com, go to the following website:

http://www.cisco.com/register/

If you cannot resolve your technical issue by using the TAC online resources, Cisco.com registered users can open a case online by using the TAC Case Open tool at the following website:

http://www.cisco.com/tac/caseopen

Contacting TAC by Telephone

If you have a priority level 1 (P1) or priority level 2 (P2) problem, contact TAC by telephone and immediately open a case. To obtain a directory of toll-free numbers for your country, go to the following website:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

P1 and P2 level problems are defined as follows:

- P1—Your production network is down, causing a critical impact to business operations if service is not restored quickly. No workaround is available.
- P2—Your production network is severely degraded, affecting significant aspects of your business operations. No workaround is available.

Using Cisco IOS Software

This chapter provides helpful tips for understanding and configuring Cisco IOS software using the command-line interface (CLI). It contains the following sections:

For an overview of Cisco IOS software configuration, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*.

For information on the conventions used in the Cisco IOS software documentation set, see the chapter "About Cisco IOS Software Documentation" located at the beginning of this book.

Understanding Command Modes

You use the CLI to access Cisco IOS software. Because the CLI is divided into many different modes, the commands available to you at any given time depend on the mode you are currently in. Entering a question mark (**?**) at the CLI prompt allows you to obtain a list of commands available for each command mode.

When you log in to the CLI, you are in user EXEC mode. User EXEC mode contains only a limited subset of commands. To have access to all commands, you must enter privileged EXEC mode, normally by using a password. From privileged EXEC mode you can issue any EXEC command—user or privileged mode—or you can enter global configuration mode. Most EXEC commands are one-time commands. For example, **show** commands show important status information, and **clear** commands clear counters or interfaces. The EXEC commands are not saved when the software reboots.

Configuration modes allow you to make changes to the running configuration. If you later save the running configuration to the startup configuration, these changed commands are stored when the software is rebooted. To enter specific configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode and a variety of other modes, such as protocol-specific modes.

ROM monitor mode is a separate mode used when the Cisco IOS software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode.

Table 1 describes how to access and exit various common command modes of the Cisco IOS software. It also shows examples of the prompts displayed for each mode.

Table 1 Accessing and Exiting Command Modes

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	`Router>`	Use the **logout** command.
Privileged EXEC	From user EXEC mode, use the **enable** EXEC command.	`Router#`	To return to user EXEC mode, use the **disable** command.
Global configuration	From privileged EXEC mode, use the **configure terminal** privileged EXEC command.	`Router(config)#`	To return to privileged EXEC mode from global configuration mode, use the **exit** or **end** command, or press **Ctrl-Z**.
Interface configuration	From global configuration mode, specify an interface using an **interface** command.	`Router(config-if)#`	To return to global configuration mode, use the **exit** command. To return to privileged EXEC mode, use the **end** command, or press **Ctrl-Z**.
ROM monitor	From privileged EXEC mode, use the **reload** EXEC command. Press the **Break** key during the first 60 seconds while the system is booting.	`>`	To exit ROM monitor mode, use the **continue** command.

For more information on command modes, refer to the "Using the Command-Line Interface" chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Getting Help

Entering a question mark (**?**) at the CLI prompt displays a list of commands available for each command mode. You can also get a list of keywords and arguments associated with any command by using the context-sensitive help feature.

To get help specific to a command mode, a command, a keyword, or an argument, use one of the following commands:

Command	Purpose
help	Provides a brief description of the help system in any command mode.
*abbreviated-command-entry***?**	Provides a list of commands that begin with a particular character string. (No space between command and question mark.)
abbreviated-command-entry<**Tab**>	Completes a partial command name.
?	Lists all commands available for a particular command mode.
command **?**	Lists the keywords or arguments that you must enter next on the command line. (Space between command and question mark.)

Example: How to Find Command Options

This section provides an example of how to display syntax for a command. The syntax can consist of optional or required keywords and arguments. To display keywords and arguments for a command, enter a question mark (**?**) at the configuration prompt or after entering part of a command followed by a space. The Cisco IOS software displays a list and brief description of available keywords and arguments. For example, if you were in global configuration mode and wanted to see all the keywords or arguments for the **arap** command, you would type **arap ?**.

The <cr> symbol in command help output stands for "carriage return." On older keyboards, the carriage return key is the Return key. On most modern keyboards, the carriage return key is the Enter key. The <cr> symbol at the end of command help output indicates that you have the option to press **Enter** to complete the command and that the arguments and keywords in the list preceding the <cr> symbol are optional. The <cr> symbol by itself indicates that no more arguments or keywords are available and that you must press **Enter** to complete the command.

Table 2 shows examples of how you can use the question mark (**?**) to assist you in entering commands. The table steps you through configuring an IP address on a serial interface on a Cisco 7206 router that is running Cisco IOS Release 12.0(3).

Table 2 How to Find Command Options

Command	Comment

`Router> enable`
`Password: <password>`
`Router#`
	Enter the **enable** command and password to access privileged EXEC commands. You are in privileged EXEC mode when the prompt changes to `Router#`.

`Router# configure terminal`
`Enter configuration commands, one per line. End with CNTL/Z.`
`Router(config)#`
	Enter the **configure terminal** privileged EXEC command to enter global configuration mode. You are in global configuration mode when the prompt changes to `Router(config)#`.

`Router(config)# interface serial ?`
`  <0-6>  Serial interface number`
`Router(config)# interface serial 4 ?`
`  /`
`Router(config)# interface serial 4/ ?`
`  <0-3>  Serial interface number`
`Router(config)# interface serial 4/0`
`Router(config-if)#`
	Enter interface configuration mode by specifying the serial interface that you want to configure using the **interface serial** global configuration command.
	
	Enter **?** to display what you must enter next on the command line. In this example, you must enter the serial interface slot number and port number, separated by a forward slash.
	
	You are in interface configuration mode when the prompt changes to `Router(config-if)#`.

Table 2 How to Find Command Options (continued)

Command	Comment

`Router(config-if)# ?`
`Interface configuration commands:`
`  .`
`  .`
`  .`
`  ip  Interface Internet Protocol config commands`
`  keepalive  Enable keepalive`
`  lan-name  LAN Name command`
`  llc2  LLC2 Interface Subcommands`
`  load-interval  Specify interval for load calculation for an interface`
`  locaddr-priority  Assign a priority group`
`  logging  Configure logging for interface`
`  loopback  Configure internal loopback on an interface`
`  mac-address  Manually set interface MAC address`
`  mls  mls router sub/interface commands`
`  mpoa  MPOA interface configuration commands`
`  mtu  Set the interface Maximum Transmission Unit (MTU)`
`  netbios  Use a defined NETBIOS access list or enable name-caching`
`  no  Negate a command or set its defaults`
`  nrzi-encoding  Enable use of NRZI encoding`
`  ntp  Configure NTP`
`  .`
`  .`
`  .`
`Router(config-if)#`
	Enter **?** to display a list of all the interface configuration commands available for the serial interface. This example shows only some of the available interface configuration commands.

`Router(config-if)# ip ?`
`Interface IP configuration subcommands:`
`  access-group  Specify access control for packets`
`  accounting  Enable IP accounting on this interface`
`  address  Set the IP address of an interface`
`  authentication  authentication subcommands`
`  bandwidth-percent  Set EIGRP bandwidth limit`
`  broadcast-address  Set the broadcast address of an interface`
`  cgmp  Enable/disable CGMP`
`  directed-broadcast  Enable forwarding of directed broadcasts`
`  dvmrp  DVMRP interface commands`
`  hello-interval  Configures IP-EIGRP hello interval`
`  helper-address  Specify a destination address for UDP broadcasts`
`  hold-time  Configures IP-EIGRP hold time`
`  .`
`  .`
`  .`
`Router(config-if)# ip`
	Enter the command that you want to configure for the interface. This example uses the **ip** command.
	
	Enter **?** to display what you must enter next on the command line. This example shows only some of the available interface IP configuration commands.

Table 2 How to Find Command Options (continued)

Command	Comment

`Router(config-if)# ip address ?`
`  A.B.C.D  IP address`
`  negotiated  IP Address negotiated over PPP`
`Router(config-if)# ip address`
	Enter the command that you want to configure for the interface. This example uses the **ip address** command.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP address or the **negotiated** keyword.
	A carriage return (<cr>) is not displayed; therefore, you must enter additional keywords or arguments to complete the command.

`Router(config-if)# ip address 172.16.0.1 ?`
`  A.B.C.D  IP subnet mask`
`Router(config-if)# ip address 172.16.0.1`
	Enter the keyword or argument you want to use. This example uses the 172.16.0.1 IP address.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP subnet mask.
	A <cr> is not displayed; therefore, you must enter additional keywords or arguments to complete the command.

`Router(config-if)# ip address 172.16.0.1 255.255.255.0 ?`
`  secondary  Make this IP address a secondary address`
`  <cr>`
`Router(config-if)# ip address 172.16.0.1 255.255.255.0`
	Enter the IP subnet mask. This example uses the 255.255.255.0 IP subnet mask.
	Enter **?** to display what you must enter next on the command line. In this example, you can enter the **secondary** keyword, or you can press **Enter**.
	A <cr> is displayed; you can press **Enter** to complete the command, or you can enter another keyword.

`Router(config-if)# ip address 172.16.0.1 255.255.255.0` `Router(config-if)#`	In this example, Enter is pressed to complete the command.

Using the no and default Forms of Commands

Almost every configuration command has a **no** form. In general, use the **no** form to disable a function. Use the command without the **no** keyword to reenable a disabled function or to enable a function that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no ip routing** command; to reenable IP routing, use the **ip routing** command. The Cisco IOS software command reference publications provide the complete syntax for the configuration commands and describe what the **no** form of a command does.

Configuration commands also can have a **default** form, which returns the command settings to the default values. Most commands are disabled by default, so in such cases using the **default** form has the same result as using the **no** form of the command. However, some commands are enabled by default and have variables set to certain default values. In these cases, the **default** form of the command enables the command and sets the variables to their default values. The Cisco IOS software command reference publications describe the effect of the **default** form of a command if the command functions differently than the **no** form.

Saving Configuration Changes

Use the **copy system:running-config nvram:startup-config** command to save your configuration changes to the startup configuration so that the changes will not be lost if the software reloads or a power outage occurs. For example:

```
Router# copy system:running-config nvram:startup-config
Building configuration...
```

It might take a minute or two to save the configuration. After the configuration has been saved, the following output appears:

```
[OK]
Router#
```

On most platforms, this task saves the configuration to NVRAM. On the Class A Flash file system platforms, this task saves the configuration to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Filtering Output from the show and more Commands

In Cisco IOS Release 12.0(1)T and later releases, you can search and filter the output of **show** and **more** commands. This functionality is useful if you need to sort through large amounts of output or if you want to exclude output that you need not see.

To use this functionality, enter a **show** or **more** command followed by the "pipe" character (|); one of the keywords **begin**, **include**, or **exclude**; and a regular expression on which you want to search or filter (the expression is case-sensitive):

command | {**begin** | **include** | **exclude**} *regular-expression*

The output matches certain lines of information in the configuration file. The following example illustrates how to use output modifiers with the **show interface** command when you want the output to include only lines in which the expression "protocol" appears:

```
Router# show interface | include protocol

FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

For more information on the search and filter functionality, refer to the "Using the Command-Line Interface" chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Identifying Platform Support for Cisco IOS Software Features

Cisco IOS software is packaged in feature sets consisting of software images intended for specific routing and switching platforms. The feature sets available for a specific hardware platform depend on which Cisco IOS software images are included in a release. Information in the following sections will help you identify the set of software images available in a specific release or to determine if a feature is available in a given Cisco IOS software image:

- Using Feature Navigator
- Using Software Release Notes

Using Feature Navigator

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a particular set of features and which features are supported in a particular Cisco IOS image.

Feature Navigator is available 24 hours a day, 7 days a week. To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, e-mail the Contact Database Administration group at cdbadmin@cisco.com. If you do not have an account on Cisco.com, go to http://www.cisco.com/register and follow the directions to establish an account.

To use Feature Navigator, you must have a JavaScript-enabled web browser such as Netscape 3.0 or later, or Internet Explorer 4.0 or later. Internet Explorer 4.0 always has JavaScript enabled. To enable JavaScript for Netscape 3.x or Netscape 4.x, follow the instructions provided with the web browser. For JavaScript support and enabling instructions for other browsers, check with the browser vendor.

Feature Navigator is updated when major Cisco IOS software releases and technology releases occur. You can access Feature Navigator at the following URL:

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

Using Software Release Notes

Cisco IOS software releases include release notes that provide the following information:

- Platform support information
- Memory recommendations
- Microcode support information
- Feature set tables
- Feature descriptions
- Open and resolved severity 1 and 2 caveats for all platforms

Release notes are intended to be release-specific for the most current release, and the information provided in these documents may not be cumulative in providing information about features that first appeared in previous releases.

Mobile Wireless Overview

A fast-paced technological transition is occurring today in the world of internetworking. This transition is marked by the convergence of the telecommunications infrastructure with that of IP data networking to provide integrated voice, video, and data services.

As this transition progresses, the standards are continuing to evolve and many new standards are being developed to enable and accelerate this convergence of telecommunications and IP networking to mobilize the internet and provide new multimedia services.

The *Cisco IOS Mobile Wireless Configuration Guide* discusses the technologies implemented in the Cisco IOS software that support mobile wireless communication and IP data services in a mobile wireless environment.

This chapter includes the following sections:

Introduction to Mobile Wireless Technology

The technologies related to wireless communication can be complex to differentiate. Wireless technology has been around for a while; however, there has been a relatively recent and rapid surge in the evolution of new wireless standards to support the convergence of voice, video and data communication. Much of this rapid evolution, or revolution, is a result of people seeking ubiquitous and immediate access to information and the assimilation of the internet into business practices and for personal use. People “on the go” want their internet access to move with them, so that their information is available at anytime, anywhere.

There are many factors that can be used to characterize wireless technologies:

- Spectrum, or the range of frequencies in which the network operates
- Transmission speeds supported
- Underlying transmission mechanism, such as frequency division multiple access (FDMA), time division multiple access (TDMA), or code division multiple access (CDMA)
- Architectural implementation, such as enterprise based (or in-building), fixed, or mobile

In addition, the mobile wireless technologies [such as Global System for Mobile Communications (GSM), TDMA, CDMA] are differentiated by a number of different factors, including some of the following:

- Control of the transmitted power
- Radio resource management and channel allocation
- Coding algorithms
- Network topology and frequency reuse
- Handoff mechanisms

The *Cisco IOS Mobile Wireless Configuration Guide* focuses on technologies that are directly related to the mobile wireless segment of wireless communication. As suggested by its name, mobile wireless communication addresses those wireless technologies that support mobility of a subscriber, which provide seamless and real-time services without interruption. Mobile wireless technologies support network access whether subscribers roam within or outside their home wireless coverage area.

Overview of Basic Network Elements Associated with Cellular Networks and Mobile Wireless

This section provides a brief introduction to a few of the basic network components associated with the existing telecommunications infrastructure. It specifically discusses the existing mobile wireless network infrastructure components for TDM-based wireless networks, some of which eventually will be replaced by new IP-based components.

In the early 1980s, support for mobile wireless communications was introduced using cellular networks, which were based on analog technologies such as AMPS. Many of the telecommunications entities associated with cellular networks still play a vital role in today's wireless networks. As wireless communications technologies continue to progress and IP data networking is further integrated into the existing infrastructure, some of the functions of these entities might still exist within the network, but will be implemented in different and more effective ways.

The following network elements are part of a typical cellular telecommunications network:

- Public Switched Telephone Network (PSTN)
- Mobile Switching Center (MSC)
- Base Station (BS)
- Radio Access Network (RAN)
- Home Location Register (HLR)
- Visitor Location Register (VLR)
- Authentication Center (AC)

Public Switched Telephone Network (PSTN)

The PSTN is the foundation and remains the predominant infrastructure that currently supports the connection of millions of subscribers worldwide. The PSTN has several thousands of miles of transmission infrastructure, including fixed land lines, microwave, and satellite links. After the introduction of cellular telephone systems in the early and mid-1980s, and with the rapid development of mobile wireless communication services, the PSTN still provides the fixed network support using the Signaling System Number 7 (SS7) protocol to carry control and signaling messages in a packet-switched environment.

Mobile Switching Center (MSC)

The MSC, usually located at the Mobile Telephone Switching Office (MTSO), is part of the mobile wireless network infrastructure that provides the following services:

- Switches voice traffic from the wireless network to the PSTN if the call is a mobile-to-landline call, or it switches to another MSC within the wireless network if the call is a mobile-to-mobile call.
- Provides telephony switching services and controls calls between telephone and data systems.
- Provides the mobility functions for the network and serves as the hub for up to as many as 100 BSs.

More specifically, the MSC provides the following functions:

- Mobility management for the subscribers (to register subscribers, to authenticate and authorize the subscribers for services and access to the network, to maintain the information on the temporary location of the subscribers so they can receive and originate voice calls).

  In GSM, some of the functionality of the MSC is distributed to the Base Station Controller (BSC). In TDMA, the BSC and the MSC are integrated.
- Call setup services (call routing based on the called number). These calls can be to another mobile subscriber through another MSC, or to a landline user through the PSTN.
- Connection control services, which determine how calls are routed and establishes trunks to carry the bearer traffic to another MSC or to the PSTN.
- Service logic functions, which route the call to the requested service for the subscriber, such as an 800 service, call forwarding, or voicemail.
- Transcoding functions, which decompress the voice traffic from the mobile device going to the PSTN and compresses the traffic going from the PSTN to the mobile device.

Base Station (BS)

The BS is the component of the mobile wireless network access infrastructure that terminates the air interface over which the subscriber traffic is transmitted to and from a mobile station (MS).

In GSM-based networks, the BS is called a Base Transceiver Station (BTS).

Radio Access Network (RAN)

The RAN identifies the portion of the wireless network that handles the radio frequencies (RF), Radio Resource Management (RRM), which involves signaling, and the data synchronization aspects of transmission over the air interface.

In GSM-based networks, the RAN typically consists of BTSs and Base Station Controllers (BSCs). User sessions are connected from a mobile station to a BTS, which connects to a BSC. The combined functions of the BTS and BSC are referred to as the Base Station Subsystem (BSS).

Home Location Register (HLR)

The HLR is a database that contains information about subscribers to a mobile network that is maintained by a particular service provider. In addition, for subscribers of a roaming partner, the HLR might contain the service profiles of visiting subscribers.

The MSC uses the subscriber information supplied by the HLR to authenticate and register the subscriber. The HLR stores “permanent” subscriber information (rather than temporary subscriber data, which a VLR manages), including the service profile, location information, and activity status of the mobile user.

Visitor Location Register (VLR)

The VLR is a database that is maintained by an MSC, to store temporary information about subscribers who roam into the coverage area of that MSC.

The VLR, which is usually part of an MSC, communicates with the HLR of the roaming subscriber to request data, and to maintain information about the subscriber's current location in the network.

Authentication Center (AC)

The AC provides handset authentication and encryption services for a service provider. In most wireless networks today, the AC is collocated with the HLR, and is often implemented as part of the HLR complex.

Wireless Standards Development

This section discusses the evolution of some of the wireless networking standards and the types of services they support.

The phased evolution of wireless networking standards are referred to as generations:

- 1G—First generation. 1G refers to the initial category of mobile wireless networks that used only analog technology and were developed primarily for voice services. Advanced Mobile Phone Service (AMPS) is an example of a 1G mobile network standard.
- 2G—Second generation. 2G refers generically to a category of mobile wireless networks and services that use digital technology. 2G wireless networks introduce support for data services. GSM, TDMA and CDMA are examples of 2G mobile network standards.
- 2.5G—Second generation plus. 2.5G (also called 2G+) refers generically to a category of mobile wireless networks that have a packet data overlay built on top of the circuit-switched voice network to support higher data rates than 2G mobile networks (2G networks support data in a circuit-switched model). General Packet Radio Service (GPRS) is an example of a 2G+ mobile network standard.

  There is a similar packet data overlay concept for CDMA called Packet Data Services Node (PDSN), but this is considered 3G as part of the CDMA 1x solution.
- 3G—Third generation. 3G refers generically to a category of next-generation mobile networks which operate at a higher frequency bandwidth (typically 2.1 GHz and higher) and have a larger channel bandwidth. This enables 3G networks to support very high data rates, up to 2 Mbps. With the higher bandwidth, more data and multimedia services are possible. 3G refers to the radio network and RF technology, and does not affect the switching core. The switching infrastructure for 3G is still based on MSCs and the TDM model.

  The Universal Mobile Telephone Service (UMTS), based on the Wideband CDMA (W-CDMA) R-99 and CDMA 2000, are examples of 3G radio networks that are being developed to fulfill the requirements in the International Mobile Telecommunications-2000 (IMT-2000) standard by the International Telecommunication Union (ITU).

- 3G+—Third generation plus. 3G+ refers to an advanced level of 3G that introduces the concept of an all-IP switching core. An all-IP switching core means that IP replaces the TDM-based MSC infrastructure with IP-based transport and IP-based signaling. IP-based signaling is implemented with new protocols like Session Initiation Protocol (SIP) and Media Gateway Control Protocol (MGCP). In 3G+ networks, the traditional MSC implementation goes away and the various MSC functions are redistributed to several other elements. A good example of this evolution in the switching core from TDM to packets is 3GPP's R4 and R5 architecture. 3GPP2 also has adopted a similar trend to transition to an all-IP network.

  There are also initiatives under way to develop and migrate to a true end-to-end, all-IP mobile wireless network where both the switching core and the RAN are IP based. This evolution is being loosely referred to as R6 in 3G terminology.

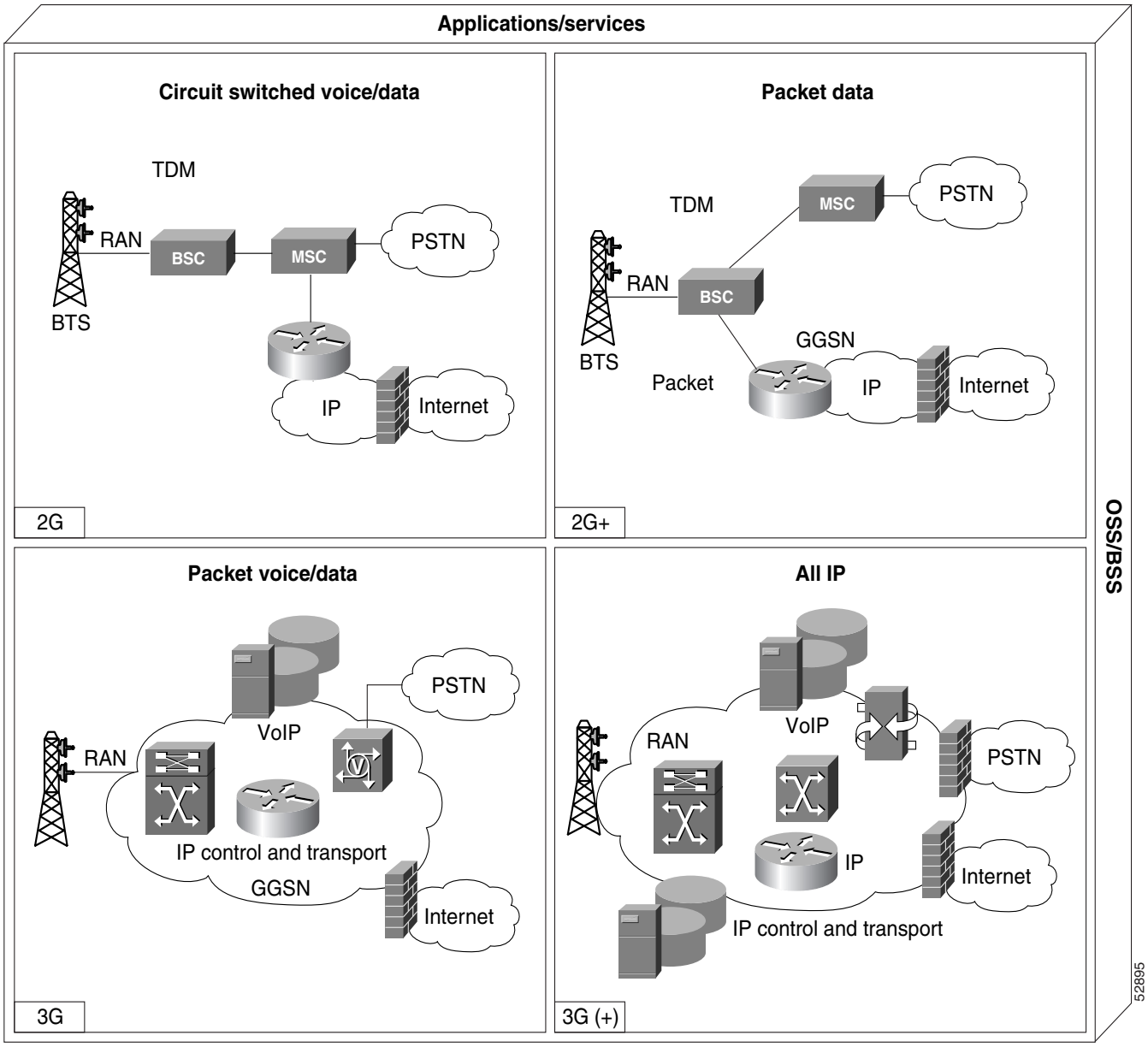
Model for IP Integration into Mobile Wireless

The standards for the integration of IP data networking with the existing telecommunications infrastructure are rapidly developing and beginning to be realized in today's production networks.

Figure 1-1 shows a model for IP integration based upon the current industry direction and reflects some of the latest ideas within the Mobile Wireless Internet Forum (MWIF). The MWIF is a pre-standards consortium for service providers and suppliers to collaborate on the implementation of IP-based mobile wireless networks. The MWIF influences the standards bodies such as 3GPP and 3GPP2 to successfully adopt new implementations.

In particular, Figure 1-1 shows where Cisco Systems' GGSN product for GSM networks fits into the model.

Figure 1-1 IP Integration Phases in Mobile Wireless



The top two quadrants in Figure 1-1 show where we are today in the telecommunications and IP data services infrastructures. The first quadrant represents the first phase of these infrastructures based on circuit-switched voice and data services. The beginnings of a core IP transport for voice and data integration can be built using Cisco Systems V.110 solutions.

The second quadrant depicts the implementation phase of 2G+ technologies, such as GPRS, supporting higher transmission speeds. In this quadrant, the Cisco Systems GGSN provides IP packet data services. It acts as an IP gateway for access to the internet and other public and private data networks for traffic that is initiated in a GSM-based mobile environment. The services anticipated in this phase include implementing always-on data services and enabling operators to charge by packet rather than connect time. Similar services are supported by Packet Data Services Node (PDSN), for CDMA-based wireless networks.

The third quadrant represents phase three of the integration of IP networking where voice and data are consolidated onto a packet-based infrastructure from the RAN or radio network control (RNC) outward. This is considered a 3G solution. Phase three enables integrated voice and data applications and reduces costs. In addition, some of the components or functions of the MSC are distributed.

The fourth quadrant represents the final phase, which includes 3G services plus the implementation of IP-based radio and mobility components to develop a true end-to-end, all-IP wireless network solution.

Mobile Wireless in Cisco IOS Software

Cisco Systems has a variety of products that provide wireless communications services and that can be used together as solutions for different network environments and needs. Some of these products provide fixed wireless IP data services and others address mobile wireless IP data services. Some reside in Cisco IOS software and others do not.

The *Cisco IOS Mobile Wireless Configuration Guide* focuses on a portion of the wireless communications services provided by the Cisco IOS software. It describes the segment of wireless products that provide *mobile* wireless communications services. This first version of the book describes a product that supports IP data services in a mobile environment.

Note The Cisco IOS software also supports the mobile IP protocol, which is not documented in this book. For more information about mobile IP, refer to the *Cisco IOS IP Configuration Guide*.

IP Data Services

This section describes the GSM-based technologies implemented in the Cisco IOS software for IP data services in mobile wireless networks—GPRS/UMTS.

GPRS/UMTS

GPRS is a service designed for GSM networks. GSM is a digital cellular technology that is used worldwide, predominantly in Europe and Asia, with current estimates of 400 million subscribers and growing. GSM is the world's leading standard in digital wireless communications.

GPRS is standardized by the European Telecommunications Standards Institute (ETSI). The most common application of GPRS is expected to be Internet/intranet access. Cisco Systems' GPRS solution enables mobile wireless service providers to supply their mobile subscribers with packet data services.

UMTS is an extension of GPRS. It is a 3G mobile communications technology that provides a range of broadband services to wireless and mobile communications. The UMTS takes a phased approach toward an all-IP network by extending 2G GPRS networks and using Wide-band Code Division Multiple Access (CDMA) technology.

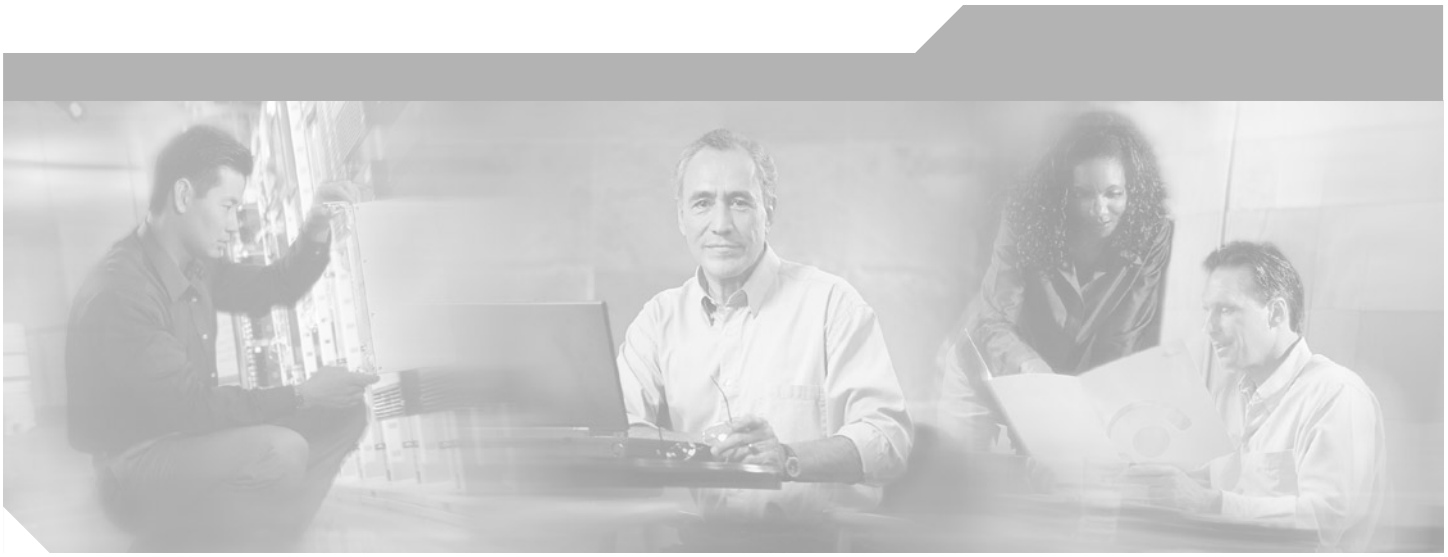
A GPRS network has two essential elements:

- Serving GPRS Support Node (SGSN)—Sends data to and receives data from mobile stations, and maintains information about the location of a mobile station (MS). The SGSN communicates between the MS and the GGSN. SGSN support is available from Cisco partners or other vendors.
- Gateway GPRS Support Node (GGSN)—A wireless gateway that allows mobile cell phone users to access the public data network (PDN) or specified private IP networks. The GGSN function is implemented on the Cisco Systems' router.

Cisco Systems is recognized as the first to market a viable GGSN product. GGSN support is available in the GPRS/UMTS feature in Cisco IOS software.

The GPRS/UMTS part of this *Cisco IOS Mobile Wireless Configuration Guide* describes how to configure a Cisco Systems router to function as a 2.5G and 3G GGSN. While the documentation provides a brief overview of the GPRS/UMTS technology and its benefits, the primary purpose of this documentation is to provide you with the necessary information to configure, verify, and monitor the GGSN portion of your GPRS/UMTS network. It does not describe all of the planning considerations that might be involved in setting up your GPRS/UMTS network.

CISCO SYSTEMS



PART 1

Gateway GPRS Support Node Release 4.0

Overview of GPRS and UMTS

This chapter provides a brief introduction to the 2.5G General Packet Radio Service (GPRS) and the 3G Universal Mobile Telecommunication System (UMTS) technologies and their implementation in Cisco IOS GGSN Release 4.0 software.

This chapter includes the following sections:

Overview

GPRS and UMTS are evolutions of the Global System for Mobile Communications (GSM) networks. GSM is a digital cellular technology that is used worldwide, predominantly in Europe and Asia. GSM is the world's leading standard in digital wireless communications.

GPRS is a 2.5G mobile communications technology that enables mobile wireless service providers to offer their mobile subscribers with packet-based data services over GSM networks. Common applications of GPRS include the following: Internet access, intranet/corporate access, instant messaging, and mutlimedia messaging. GPRS was standardized by the European Telecommunications Standards Institute (ETSI), but today is standardized by the Third Generation Partnership Program (3GPP).

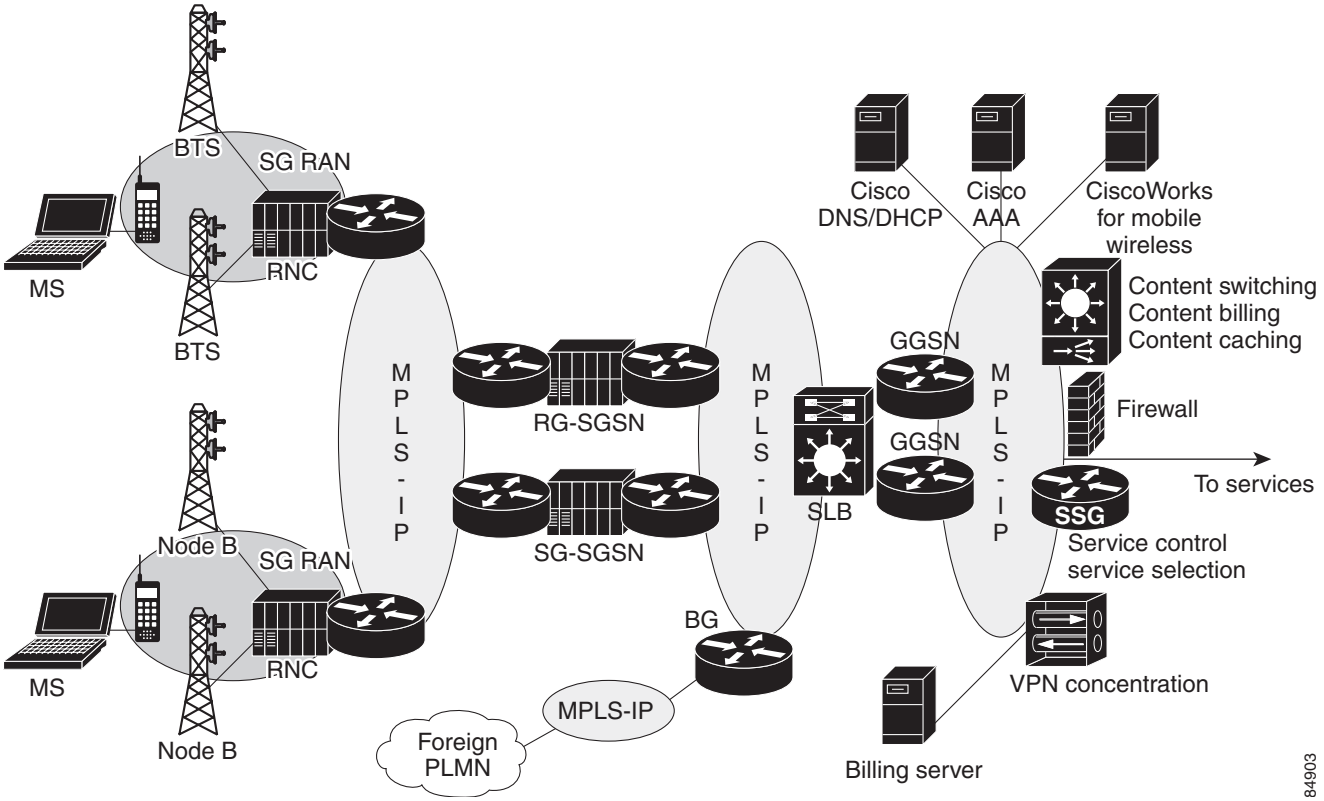
UMTS is a 3G mobile communications technology that provides Wide-band Code Division Multiple Access (CDMA) radio technology. The CDMA technology offers higher throughput, real-time services, and end-to-end QoS, and is designed to deliver pictures, graphics, video communications, and other multimedia information as well as voice and data to mobile wireless subscribers. UMTS is standardized by the Third Generation Partnership Program (3GPP).

The GPRS/UMTS packet core is primarily composed of two major network elements:

- Gateway GPRS Support Node (GGSN)—A gateway that provides mobile cell phone users access to a public data network (PDN) or specified private IP networks. The GGSN function is implemented on the Cisco Systems' router via Cisco IOS software. The Cisco IOS GGSN 4.0 feature provides both the 2.5G GPRS and 3G UMTS GGSN functions.
- Serving GPRS Support Node (SGSN)—Connects the Radio Access Network (RAN) to the GPRS/UMTS core and tunnels user sessions to the GGSN. The SGSN sends data to and receives data from mobile stations, and maintains information about the location of a mobile station (MS). The SGSN communicates directly with the MS and the GGSN. SGSN support is available from Cisco partners or other vendors.

Figure 2-1 provides a view of the basic GPRS/UMTS network components.

Figure 2-1 GPRS/UMTS Network Components



Note that, as illustrated in Figure 2-1, the Radio Access Network (RAN) is made up of different components for 2.5G and 3G.

In a 2.5G environment, the RAN is comprised of mobile stations that connect to a Base Transceiver Station (BTS) that connects to a Base Station Controller (BSC). In a 3G environment, the RAN is made up of mobile stations that connect to a NodeB that connects to a Radio Network Controller (RNC).

The RAN then connects to the GPRS/UMTS core through an SGSN, which tunnels user sessions to a GGSN that act as a gateway to the services networks (for example, the Internet and intranet). The connection between the SGSN and the GGSN is enabled through a tunneling protocol called the GPRS Tunneling Protocol (GTP); GTP Version 0 (GTP V0) for 2.5G applications and GTP Version 1 (GTP V1) for 3G applications. GTP is carried over IP. Multiple SGSNs and GGSNs within a network are referred to collectively as GPRS Support Nodes (GSNs).

Note Depending on the specific operator configuration, the RAN, GPRS/UMTS core, and the services networks can be made up of IP or MPLS networks.

To assign mobile sessions an IP address, the GGSN uses the Dynamic Host Configuration Protocol (DHCP). The GGSN can use a Remote Authentication Dial-In User Service (RADIUS) server to authorize and authenticate the remote users. DHCP and RADIUS services can be specified at the global configuration level (using GPRS DHCP and RADIUS commands), or for each access point configured on the GGSN.

In Cisco IOS Release 12.1(5)T and later, the GGSN (with an Industry-Standard Architecture [ISA] card), supports the IP security protocol (IPSec) to provide data confidentiality, data integrity, and data authentication between participating peers.

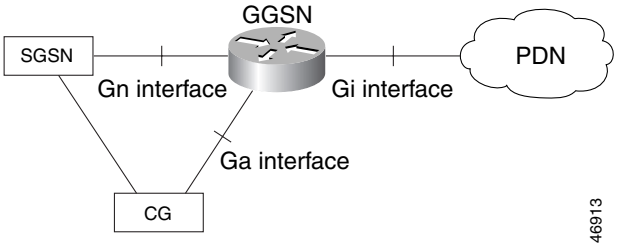
GPRS Interface Reference Model

The 2.5G GPRS and 3G UMTS standards use the term *interface* to label (or identify) the communication path between different network elements. The GPRS/UMTS standards define the requirements and characteristics of communication between different GPRS/UMTS network elements over these interfaces. These interfaces are commonly referred to when describing aspects of a GPRS/UMTS network.

Figure 2-2 shows the interfaces that are implemented in the Cisco IOS GGSN Release 4.0 feature:

- Gn interface—Interface between GSNs within the same PLMN in a GPRS/UMTS network. GTP is a protocol defined on the Gn interface between GSNs in a GPRS/UMTS network.
- Gi interface—Reference point between a GPRS/UMTS network and an external packet data network.
- Ga interface—Interface between a GGSN and charging gateway (CG) in a GPRS/UMTS network.

Figure 2-2 GPRS Interfaces Implemented in the Cisco IOS GGSN Feature



Virtual Template Interface

To facilitate configuration of connections between the GGSN and SGSN, and the GGSN and PDNs, the Cisco IOS GGSN software uses an internal interface called a virtual template interface. A virtual template is a logical interface on the router. A logical interface configuration on the router is not tied directly to a specific physical interface, but it can be associated dynamically with a physical interface.

As with a physical interface on the router, you can assign an IP address to the virtual template interface. You can also configure IP routing characteristics on the virtual template interface. You are required to configure certain GPRS/UMTS-specific elements on the virtual template interface, such as GTP encapsulation (which is necessary to communicate with the SGSN) and the access list that the GGSN uses to determine which PDNs are accessible on the network.

Access Points

The GPRS/UMTS standards define a network identity called an access point name (APN). An APN identifies the service or network to which a user a user can connect to from a GGSN in a GPRS/UMTS network.

To configure APNs, the Cisco IOS GGSN software uses the following configuration elements:

- Access point—Defines an APN and its associated access characteristics, including security and method of dynamic addressing.
- Access point list—Logical interface that is associated with the virtual template of the GGSN. The access-point list contains one or more access points.
- Access group—An additional level of security on the router that is configured at an access point to control access to and from a PDN. When an MS is permitted access to the GGSN as defined by a traditional IP access list, the IP access group further defines whether access is permitted to the PDN (at the access point). The IP access group configuration can also define whether access from a PDN to an MS is permitted.

For more detailed information on access-point configuration, refer to the "Configuring Access Points on the GGSN" section on page 6-6 in the "Configuring Network Access to the GGSN" chapter.

Benefits

The 2.5 G GPRS technology provides the following benefits:

- Enables the use of a packet-based air interface over the existing circuit-switched GSM network, which allows greater efficiency in the radio spectrum because the radio bandwidth is used only when packets are sent or received.
- Supports minimal upgrades to the existing GSM network infrastructure for those network service providers who want to add GPRS services on top of GSM, which is currently widely deployed.
- Supports enhanced data rates compared to the traditional circuit-switched GSM data service.
- Supports larger message lengths than Short Message Services (SMS).
- Supports a wide range of access to data networks and services, including virtual private network (VPN)/Internet service provider (ISP) corporate site access and Wireless Application Protocol (WAP).

In addition to the above, the 3G UMTS technology extends these benefits to include:

- Enhanced data rates of approximately:
  - 144 kbps—Satellite and rural outdoor
  - 384 kbps—Urban outdoor
  - 2048 kbps—Indoor and low range outdoor.
- Supports connection-oriented Radio Access Bearers with specified QoS, enabling end-to-end QoS.

The Cisco IOS GGSN 4.0 feature is a fully-compliant 2.5G and 3G GGSN that provides the following:

- Release 99 (R99), Release 98 (R98), and Release 97 (R97) support and compliance
- GTP v0 and GTP v1 messaging
- UMTS QoS support
- GPRS QoS (R97/R98) conversion to UMTS QoS (R99) and the reverse
- R99 charging
- GGSN interworking between 2.5G and 3G SGSNs with RA update from:
  - 2.5G to 2.5G SGSN
  - 2.5G to 3G SGSN

  - 3G to 3G SGSN
  - 3G to 2.5G SGSN
- R97/R98 Ga support
- R99 Ga support
- 2.5G and 3G MIB support

Planning to Configure the GGSN

This chapter describes information that you should know before configuring the GGSN.

This chapter includes the following sections:

Prerequisites

Planning Your Access Point Configuration

Before you begin to configure the GGSN on your router, you should know which networks your mobile users will be allowed to access using the GGSN. Once you identify the networks, you can plan the physical interfaces to configure on the router for those networks. Then you can plan the associated access points to those networks and configure them on the GGSN.

For example, you might want to provide user access to the World Wide Web through a PDN, plus access to two private corporate intranets. In this case, you need to set up three access points—one to enable user access to the PDN, and one for each private intranet.

Restrictions

The number of PDP contexts supported on the GGSN is dependent on the memory and router series in use, your GGSN configuration (whether a method of Point to Point Protocol [PPP] has been configured to forward packets beyond the terminal equipment and mobile termination), and the rate of PDP context creation to be supported.

The following list shows the maximum number of PDP contexts supported on the GGSN according to the memory and router series in use when a method of PPP has not been configured:

- Cisco 7206 VXR NPE-300 router with 256 Mb RAM—80,000 PDP contexts.
- Cisco 7206 VXR NPE-400 router with 512 Mb RAM—135,000 PDP contexts.

For information on the maximum number of PDP contexts supported on the GGSN when a method of PPP has been configured, see “Overview of PPP Support on the GGSN” section on page 7-1.

Supported Platforms

- Cisco 7200 VXR NPE-300
- Cisco 7200 VXR NPE-400

Supported Standards, MIBs, and RFCs

Standards

Cisco IOS GGSN software release 4.0 supports the following 3GPP standards:

- Release 97/98
  - 3G TS 03.03
  - 3G TS 03.60
  - 3G TS 04.08
  - 3G TS 09.02
  - 3G TS 09.60
  - 3G TS 09.61
  - 3G TS 12.15
- Release 99
  - 3G TS 22.107
  - 3G TS 23.003
  - 3G TS 23.060
  - 3G TS 24.008
  - 3G TS 29.002
  - 3G TS 29.060
  - 3G TS 29.061
  - 3G TS 32.015

The GGSN interfaces comply with the following SMG (Special Mobile Group) standards:

- Ga interface—SMG#28 R99
- Gn interface—SMG#31 R98

MIBs

- CISCO-GGSN-MIB
- CISCO-GGSN-QOS-MIB
- CISCO-GPRS-ACC-PT-MIB
- CISCO-GPRS-CHARGING-MIB
- CISCO-GPRS-GTP-CAPABILITY
- CISCO-GPRS-GTP-MIB
- CISCO-GTP-CAPABILITY

- CISCO-GTP-MIB
- CISCO-GTP-DIRECTOR_MIB

Note The CISCO-GPRS-GTP-CAPABILITY MIB describes the scope of objects supported in the CISCO-GPRS-GTP-MIB. The CISCO-GTP-CAPABILITY MIB describes the scope of objects supported in the CISCO-GTP-MIB.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

- RFC 1518, *An Architecture for IP Address Allocation with CIDR*
- RFC 1519, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*
- RFC 1661, *The Point-to-Point Protocol (PPP)*
- RFC 2475, *An Architecture for Differentiated Services*

Related Documents

- *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2
- *Cisco IOS Dial Technologies Command Reference*, Release 12.2
- *Cisco IOS Interface Configuration Guide*, Release 12.2
- *Cisco IOS Interface Command Reference*, Release 12.2
- *Cisco IOS IP Configuration Guide*, Release 12.2
- *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services*, Release 12.2
- *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*, Release 12.2
- *Cisco IOS IP Command Reference, Volume 3 of 3: Multicast*, Release 12.2
- *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2
- *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.2
- *Cisco IOS Security Configuration Guide*, Release 12.2
- *Cisco IOS Security Command Reference*, Release 12.2
- *Cisco IOS Switching Services Configuration Guide*, Release 12.2
- *Cisco IOS Switching Services Command Reference*, Release 12.2

Configuring GGSN GTP Services

This chapter describes how to configure a Cisco router as a GGSN, and how to configure GTP options.

For a complete description of the GGSN commands in this chapter, refer to the *Cisco IOS Mobile Wireless Command Reference*.

To locate documentation of other commands that appear in this chapter, use the command reference master index or search online. See the "Related Documents" section on page 3-3 for a list of the other Cisco IOS software documentation that might be helpful while configuring the GGSN.

This chapter includes the following sections:

GTP Overview

GTP is the protocol used to tunnel multi-protocol packets through the GPRS/UMTS network. It is defined on the Gn interface as the protocol between GSNs in the GPRS/UMTS backbone network.

With GGSN 4.0, the Cisco GGSN supports both GTP Version 0 (GTP v0) and GTP Version 1 (GTP v1) simultaneously. GPRS R97/R98 uses GTP Version 0 and UMTS R99 uses GTP v1.

The GGSN automatically selects the GTP version to use according to the capabilities of the SGSN. In the case of a network-initiated PDP activation, the GGSN sends a PDU Notification Request message with GTP v1. If an ICMP error messages is received, GGSN immediately resends the message with GTP v0.

Configuring the Router for GGSN Services

The Cisco IOS GGSN software uses a logical interface called a virtual template interface to configure the router as a GGSN. This section describes the primary tasks you need to complete when configuring the router for GGSN services. Once the router has been configured as a GGSN, the subsequent configuration tasks describe how to establish connectivity from the GGSN to the SGSN and PDNs.

The following requirements must be met when configuring the GGSN on a Cisco router:

- Configure only a single GGSN entity on each router using the **service gprs ggsn** global configuration command.
- Configure only a single virtual template interface (as virtual template number 1) with GTP encapsulation on the GGSN.

GGSN Services Configuration Task List

To configure the router for GGSN services, perform the following tasks:

- Enabling GGSN Services, page 4-2
- Creating a Loopback Interface, page 4-2
- Creating a Virtual Template Interface for GGSN, page 4-3

Enabling GGSN Services

Configure only a single GGSN entity on each router using the **service gprs ggsn** global configuration command.

To enable GGSN services on the router, use the following command in global configuration mode:

Command	Purpose
`Router(config)# service gprs ggsn`	Specifies that the router functions as a GGSN.

Creating a Loopback Interface

Rather than directly configuring an IP address on the virtual template, Cisco recommends that you create a loopback interface and then associate the loopback interface IP address to the virtual template used for GTP encapsulation. Cisco recommends that you configure the virtual template interface as unnumbered, and associate its IP numbering with a loopback interface.

A loopback interface is a software-only interface that emulates an interface that is always up. It is a virtual interface supported on all platforms. The interface-number is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces you can create. The GGSN uses loopback interfaces to support the configuration of several different features.

To create a loopback interface, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface loopback number`	Creates a loopback interface. A loopback interface is a virtual interface that is always up.
Step 2	`Router(config-if)# ip address ip-address mask`	Assigns an IP address to the loopback interface.

Creating a Virtual Template Interface for GGSN

Configure only a single virtual template interface (as virtual template number 1) with GTP encapsulation on the GGSN.

To create a virtual template interface for GGSN, use the following command beginning in global configuration mode:

	Command	Purpose
Step 3	`Router(config)# interface virtual-template number`	Creates a virtual template interface, where *number* identifies the virtual template interface. This command enters you into interface configuration mode. **Note** The GGSN supports only a single virtual template for the GTP virtual interface.
Step 4	`Router(config-if)# ip unnumber loopback number`	Assigns the previously defined loopback IP address to the virtual template interface.
Step 5	`Router(config-if)# encapsulation gtp`	Specifies GTP as the encapsulation type for packets transmitted over the virtual template interface.
Step 6		

Configuring Echo Timing on the GGSN

The GGSN uses echo timing to determine whether an SGSN or external charging gateway is active.

For a GTP path to be active, the SGSN needs to be active. To determine that an SGSN is active, the GGSN and SGSN exchange echo messages. Although the GGSN supports different methods of echo message timing, the basic echo flow begins when the GGSN sends an echo request message to the SGSN. The SGSN sends a corresponding echo response message back to the GGSN.

If the GGSN does not receive a response after a certain number of retries (a configurable value), the GGSN assumes that the SGSN is not active. This indicates a GTP path failure, and the GGSN clears all PDP context requests associated with that path.

This section describes the different methods of echo timing that are supported on the GGSN and how to configure them. It includes the following topics:

Overview of the Echo Timing Methods on the GGSN

The GGSN supports two different methods of echo timing—the default echo timer and the dynamic echo timer. Only a single method can be in use at any time on the GGSN. The following sections describe these two methods:

Note For simplicity, this document describes the operation of echo timing between the GGSN and an SGSN. If an external charging gateway is in use in the GPRS/UMTS network, the GGSN uses the same echo timing methods to maintain the charging gateway path.

Overview of the Default Echo Timer

The default echo timer is enabled on the GGSN automatically. However, you can choose to enable the dynamic echo timing method as an alternative.

When you are using the default echo timer on the GGSN, the following commands apply:

- **gprs gtp n3-requests**—Specifies the maximum number of times that the GGSN attempts to send a echo-request message. The default is 5 times.
- **gprs gtp path-echo-interval**—Specifies the number of seconds that the GGSN waits for a response from an SGSN or external charging gateway, and after receiving a response, the number of second the GGSN waits before sending the next echo-request message. The default is 60 seconds.
- **gprs gtp t3-response**—Specifies the the initial number of seconds that the GGSN waits before resending a signaling request message when a response to a request has not been received. This time is doubled for every retry. The default is 1 second.

Figure 4-1 shows the default echo request sequence when a response is successfully received within the specified path echo interval. If the GGSN receives the echo response within the path echo interval (as specified in the **gprs gtp path-echo-interval** command; default is 60 seconds), it sends another echo request message after 60 seconds (or whatever time was configured in the **gprs gtp path-echo-interval** command). This message flow continues as long as the GGSN receives an echo response message from the SGSN within the specified path echo interval.

Figure 4-1 Default GTP Path Echo Interval Request Sequence in Path Success Mode

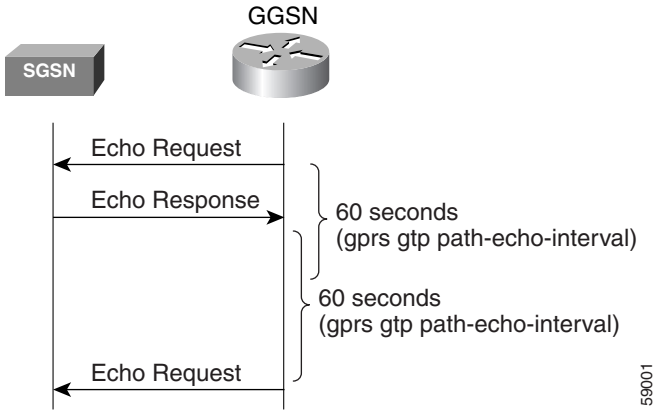
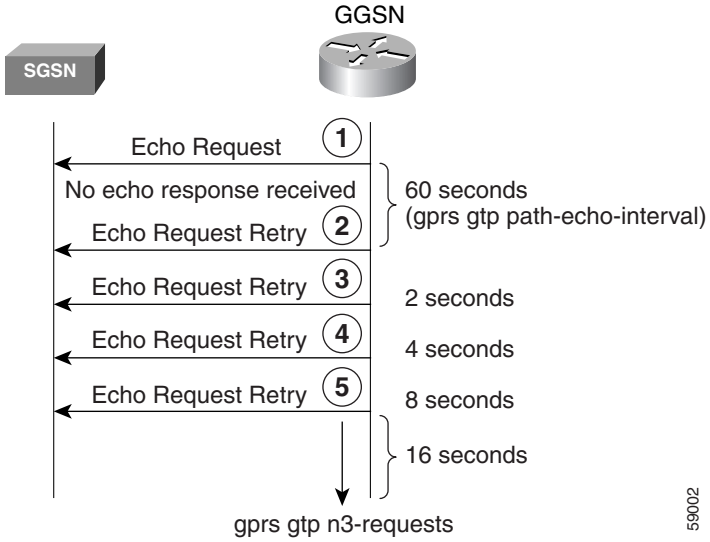


Figure 4-2 shows the default echo request sequence when the GGSN fails to receive a response to its echo request within the specified path echo interval. If the GGSN fails to receive an echo response message from the SGSN within the path echo interval, it resends echo request messages until the N3-requests counter is reached (as specified by the **gprs gtp n3-requests** command; default is 5). Because the initial request message is included in the N3-requests counter, the total number of retries is N3-1. The T3 timer increases by a factor of two for each retry (the factor value is not configurable).

Figure 4-2 Default Echo Timing Request Sequence in Path Failure Mode



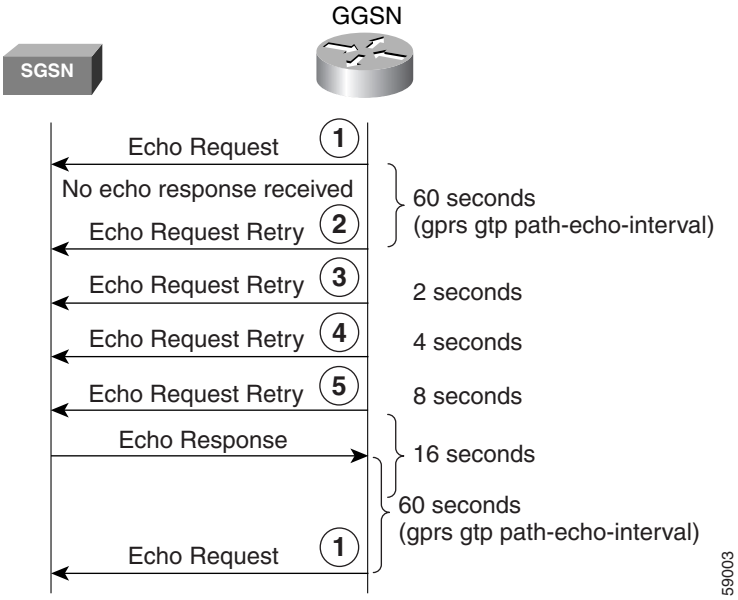
For example, if N3 is set to the default of 5, and T3 is set to the default of 1 second, the GGSN will resend 4 echo request messages (the initial request + 4 retries=5). If the GGSN does not receive an echo response from the SGSN during the 60-second path echo interval, then the GGSN immediately sends the first echo request retry message at the expiration of the path echo interval. The T3 time increments for each additional echo request, by a factor of 2 seconds, as long as the GGSN does not receive an echo response. So, the GGSN resends another message in 2 seconds, 4 seconds, and 8 seconds. After the 5th message, the GGSN waits for a final period of 16 seconds for an echo response.

If the GGSN fails to receive an echo response message from the SGSN within the time period of the N3-requests counter, it deletes all of the PDP contexts and clears the GTP path. For this example, the total elapsed time from when the first request message is sent to when PDP contexts are cleared, is: 60+2+4+8+16=90 seconds, where 60 is the initial value of the path echo interval, and the remaining 4 time periods are the increments of the T3 timer for the subsequent retries. The path is cleared after another 60- second period, or 150 seconds.

If the GGSN receives an echo response within the N3 x T3 transmission period, it goes back to success mode for its echo request sequences.

Figure 4-3 shows the GGSN receiving an echo response message within N3 x T3 retransmissions of an echo request. In this scenario, the GGSN sent an initial echo request followed by 4 retries for a total of 5 requests, according to the default setting of 5 N3 requests. The GGSN receives the echo response after the 5th and final retry, within the remaining 16 seconds. Now the GGSN is back in success mode, and it waits 60 seconds (the value of the **gprs gtp path-echo-interval** command) before sending the next echo request message.

Figure 4-3 Default Echo Timing with Echo Response Received Within N3 x T3 Retransmissions



Overview of the Dynamic echo timer

The GGSN's default echo timer cannot be configured to accommodate network congestion and, therefore, the GTP path could be cleared prematurely. The dynamic echo timer feature enables the GGSN to better manage the GTP path during periods of network congestion. Use the **gprs gtp echo-timer dynamic enable** command to enable the GGSN to perform dynamic echo timing.

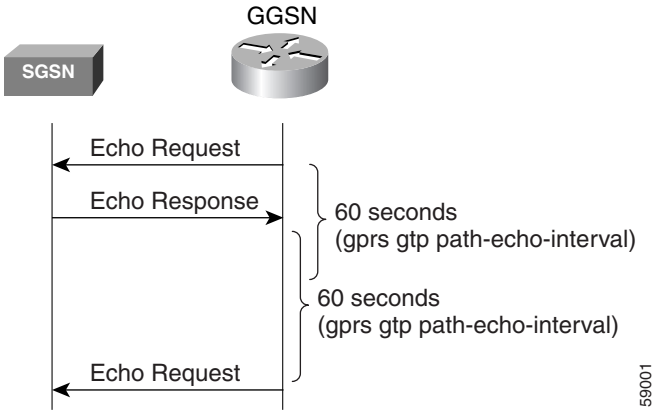
The dynamic echo timer method is different from the default echo timer method on the GGSN because it uses a calculated round-trip timer (RTT), as well as a configurable factor or multiplier to be applied to the RTT statistic. Different paths can have a different RTT, so the dynamic echo timer can vary for different paths.

When you are using the dynamic echo timer on the GGSN, the following commands apply:

- **gprs gtp echo-timer dynamic enable**—Enables the dynamic echo timer on the GGSN.
- **gprs gtp echo-timer dynamic minimum**—Specifies the minimum time period (in seconds) for the dynamic echo timer. If the RTT multiplied by the smooth factor is less than this value, the GGSN uses the value set in this command. The default is 5 seconds.
- **gprs gtp echo-timer dynamic smooth-factor**—Specifies the multiplier that the dynamic echo timer uses when calculating the time to wait to send retries, when it has not received a response from the SGSN within the path echo interval. The default is 2.
- **gprs gtp n3-requests**—Specifies the maximum number of times that the GGSN attempts to send an echo-request message. The default is 5 times.
- **gprs gtp path-echo-interval**—Specifies the number of seconds that the GGSN waits, after receiving a response from an SGSN or external charging gateway, before sending the next echo-request message. The default is 60 seconds.

Figure 4-4 shows the dynamic echo request sequence when a response is successfully received within the specified path echo interval. Just as in the default echo timing method, if the GGSN receives the echo response within the path echo interval (as specified in the **gprs gtp path-echo-interval** command; default is 60 seconds), it sends another echo request message after 60 seconds (or whatever time was configured in the **gprs gtp path-echo-interval** command). This message flow continues as long as the GGSN receives an echo response message from the SGSN within the specified path echo interval.

Figure 4-4 Dynamic GTP Path Echo Interval Request Sequence in Path Success Mode



The GGSN calculates the RTT statistic for use by the dynamic echo timer feature. The RTT is the amount of time between sending a particular echo request message and receiving the corresponding echo response message. RTT is calculated for the first echo response received (see Figure 4-5); the GGSN records this statistic. Because the RTT value might be a very small number, there is a minimum time for the dynamic echo timer to use. This value is configured using the **gprs gtp echo-timer dynamic minimum** command.

Figure 4-5 Dynamic Echo Timing Request Sequence RTT Calculation

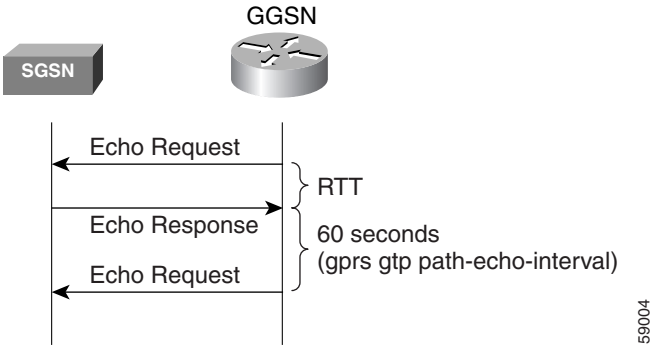
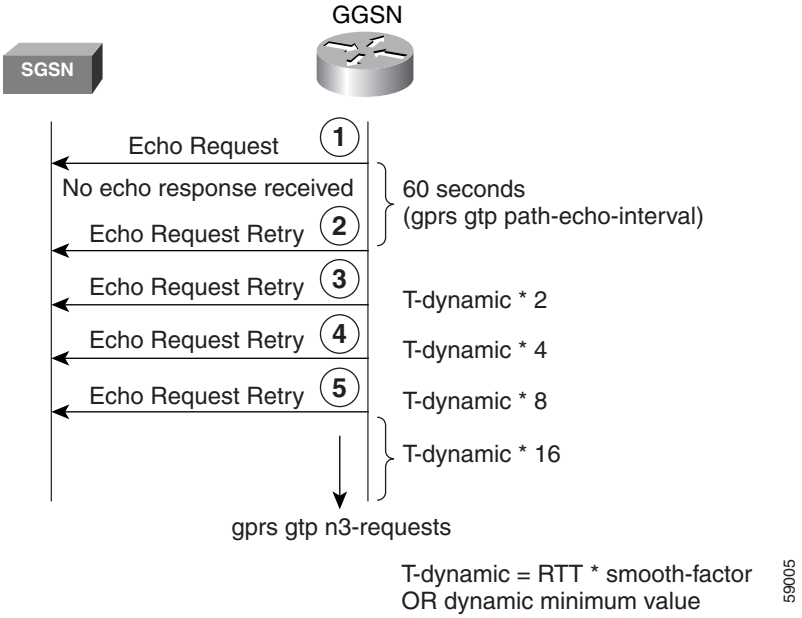


Figure 4-6 shows the dynamic echo timing request sequence in path failure mode. If the GGSN fails to receive an echo response message from the SGSN within the path echo interval, it goes into retransmission, or path failure mode. During path failure mode, the GGSN uses a value referred to as the T-dynamic. The T-dynamic is the greater of either the dynamic minimum, or the RTT statistic multiplied by the smooth factor.

Figure 4-6 Dynamic Echo Timing Request Sequence in Path Failure Mode



The T-dynamic essentially replaces the use of the **gprs gtp t3-response** command, which is used in the default echo timer method on the GGSN. The T-dynamic timer increases by a factor of two for each retry (again, this factor is not configurable), until the N3-requests counter is reached (N3-requests counter includes the initial request message).

For example, if the RTT is 6 seconds, the dynamic minimum is 5 seconds, N3 is set to 5, and the smooth factor is set to 3, the GGSN will resend up to 4 echo request messages (the initial request + 4 retries=5) in path failure mode. If the GGSN does not receive an echo response from the SGSN during the 60-second path echo interval, then the GGSN immediately sends the first echo request retry message at the expiration of the path echo interval. The RTT x smooth factor equals 18 seconds (6 x 3), which is greater than the dynamic minimum of 5 seconds, so the dynamic minimum value is not used. The

T-dynamic value is 18 (RTT x smooth factor), so the GGSN sends another retry echo request message in 36 seconds (18 x 2), 72 seconds (18 x 4), and 144 seconds (18 x 8). After the 5th message, the GGSN waits for a final period of 288 seconds (18 x 16) for an echo response.

If the GGSN fails to receive an echo response message from the SGSN in this time period, it clears the GTP path and deletes all PDP contexts. The total elapsed time from when the first request message is sent, to when the PDP contexts are cleared is:

60+36+72+144+288=600 seconds,

where 60 is the initial value of the path echo interval, and the remaining 4 time periods are the increments of the T-dynamic for the subsequent retries. The path is cleared after another 60-second period, or 660 seconds.

If the GGSN receives an echo response within the N3 x T-dynamic transmission period, it goes back to success mode for its echo request sequences. In success mode, the GGSN begins echo requests and awaits responses according to the specified path echo interval as shown in Figure 4-4.

Sequence Numbering for Retransmissions

The GGSN does not increment the sequence number of an echo request message during retransmissions. Therefore, during the period when an echo response has not been received by the GGSN, the GGSN continues to use the same sequence number for all echo request retries until the N3 requests limit has been reached, or until a response has been received. When a response is received, the sequence number of the next echo request message is incremented by 1.

If the GGSN has sent an echo request message with a higher sequence number, but still receives echo responses for sequence numbers lower than the current echo request message, the response is ignored.

Echo Timing Configuration Task List

This section describes the tasks required to customize the default echo timing method, or to enable and configure the dynamic echo timing method on the GGSN. By default, the GGSN activates the default echo timing method.

To configure echo timing on the GGSN, perform the following tasks:

- Customizing the Default Echo Timer, page 4-10 (Recommended, if used)
- Configuring the Dynamic Echo Timer, page 4-10 (Optional)
- Disabling the Echo Timer, page 4-11 (Optional)

Customizing the Default Echo Timer

The default echo timing method is enabled automatically on the GGSN. If you want to use the default echo timer, Cisco Systems recommends that you modify the following commands to optimize your network as necessary.

To customize the default echo timing method on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gprs gtp n3-requests requests`	(Optional) Specifies the maximum number of times that the GGSN attempts to send a signaling request to an SGSN. The default is 5.
Step 2	`Router(config)# gprs gtp path-echo-interval interval`	(Optional) Specifies the number of seconds that the GGSN waits, after receiving a response from an SGSN or external charging gateway, before sending the next echo-request message. The default is 60 seconds.
Step 3	`Router(config)# gprs gtp t3-response response-interval`	(Optional) Specifies the the initial time that the GGSN waits before resending a signaling request message when a response to a request has not been received. This time is doubled for every retry. The default is 1 second.

Configuring the Dynamic Echo Timer

To activate the dynamic echo timing method on the GGSN, you must enable the dynamic echo timer. After you activate the dynamic echo timer, you can modify the corresponding options to optimize the timing parameters for your network.

To configure the dynamic echo timing method on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gprs gtp echo-timer dynamic enable`	Enables the dynamic echo timer on the GGSN.
Step 2	`Router(config)# gprs gtp echo-timer dynamic minimum number`	(Optional) Specifies the minimum time period used by the dynamic echo timer. The default is 5 seconds.
Step 3	`Router(config)# gprs gtp echo-timer dynamic smooth-factor number`	(Optional) Specifies the multiplier that the GGSN uses to calculate the time to wait to send retries of the dynamic echo timer. The default is 2.
Step 4	`Router(config)# gprs gtp n3-requests requests`	(Optional) Specifies the maximum number of times that the GGSN attempts to send a signaling request to an SGSN. The default is 5.
Step 5	`Router(config)# gprs gtp path-echo-interval interval`	(Optional) Specifies the number of seconds that the GGSN waits, after receiving a response from an SGSN or external charging gateway, before sending the next echo-request message. The default is 60 seconds.

Disabling the Echo Timer

If for some reason you need to disable the GGSN from performing echo processing with an SGSN or external charging gateway, you can specify 0 seconds for the path echo interval.

To disable the echo timer, use the following command in global configuration mode:

Command	Purpose
`Router(config)# gprs gtp path-echo-interval 0`	(Optional) Specifies a path interval of 0 seconds, which disables the GGSN from performing echo processing.

Verifying the Echo Timing Configuration

This section describes how to verify the echo timing method on the GGSN. It includes the following topics:

- Verifying Echo Timing Parameters, page 4-11
- Verifying the Dynamic Echo Timer by GTP Path, page 4-12

Verifying Echo Timing Parameters

To verify the parameters in use by the GGSN for echo timing, you can use the **show gprs gtp parameters** or **show running-config** privileged EXEC commands.

The GGSN automatically sets default values for those parameters applicable to the dynamic echo timer, even when the dynamic echo timer is not enabled. Therefore, the **show gprs gtp parameters** command does not indicate which echo timing method is currently activated.

Verifying Default Echo Timing Parameters

To verify the parameters in use by the default echo timer, use the **show gprs gtp parameters** privileged EXEC command and observe the following parameters highlighted in bold below:

```
Router# show gprs gtp parameters
    GTP path echo interval                        = 60
    GTP signal max wait time T3_response          = 1
    GTP max retry N3_request                      = 5
    GTP dynamic echo-timer minimum                = 5
    GTP dynamic echo-timer smooth factor          = 2
    GTP buffer size for receiving N3_buffer       = 8192
    GTP max pdp context                           = 45000
    GPRS MCC Code                                 = 310
    GPRS MNC Code                                 = 15
```

Verifying Dynamic Echo Timing Parameters

To verify the parameters in use by the dynamic echo timer, use the **show gprs gtp parameters** privileged EXEC command and observe the parameters highlighted in bold below:

```
Router# show gprs gtp parameters
    GTP path echo interval                        = 60
    GTP signal max wait time T3_response          = 1
    GTP max retry N3_request                      = 5
    GTP dynamic echo-timer minimum                = 5
    GTP dynamic echo-timer smooth factor          = 2
    GTP buffer size for receiving N3_buffer       = 8192
    GTP max pdp context                           = 45000
    GPRS MCC Code                                 = 310
    GPRS MNC Code                                 = 15
```

Verifying the Dynamic Echo Timer by GTP Path

You can use the **show running-config** privileged EXEC command to verify whether the dynamic echo timer is enabled.

The value of the dynamic echo timer varies for each GTP path on the GGSN. To verify whether the dynamic echo timer is enabled on the GGSN, and to verify the value (in seconds) of the dynamic echo timer (T-dynamic), use the **show gprs gtp path** privileged EXEC command.

If the dynamic echo timer is not activated, the word “Disabled” appears beside the corresponding path in the Dynamic echo timer output field.

Step 1 To verify that the dynamic echo timer is enabled, use the **show running-config** command and verify that the **gprs gtp dynamic echo-timer enable** command appears as shown in bold toward the end of the following sample output:

```
Router# show running-config

Current configuration : 6769 bytes
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service gprs ggsn
!
. . .
!

interface loopback 1
 ip address 10.41.41.1 255.255.255.0
!!
interface Virtual-Template1
 ip unnumber loopback 1
 encapsulation gtp
 gprs access-point-list gprs
!
. . .
!
gprs access-point-list gprs
  access-point 1
   access-point-name gprs.cisco.com
   exit
```

```
   !
  access-point 2
   access-point-name gprt.cisco.com
   access-mode non-transparent
   aaa-group authentication test2
   aaa-group accounting test2
   ip-address-pool dhcp-proxy-client
   dhcp-server 10.65.0.1
   dhcp-gateway-address 10.65.0.1
   exit
   !
!
gprs ms-address exclude-range 10.21.1.0 10.21.1.5
gprs gtp echo-timer dynamic enable
gprs gtp echo-timer dynamic smooth-factor 5
gprs gtp echo-timer dynamic minimum 10
gprs gtp response-message wait-accounting
!
. . .
!
end
```

Step 2 To verify the T-dynamic values for the corresponding GTP paths, use the **show gprs gtp path all** privileged EXEC command.

The following example indicates that the dynamic echo timer is enabled on the GGSN, and shows that the T-dynamic values of 5 seconds and 2 seconds are in use for the corresponding paths:

```
Router# show gprs gtp path all
         Total number of path : 2

Local address           Remote address          GTP version    Dynamic echo timer
10.41.41.1(3386)        10.18.18.200(3386)      0              5
10.10.10.1(2123)        10.10.10.4(2123)        1              2
```

Customizing the GGSN Configuration

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

For information about configuring GPRS/UMTS charging options, see the "Customizing the Charging Gateway" section on page 5-6 in the "Configuring Charging on the GGSN" chapter.

This section includes the following topics:

Configuring GTP Signaling Options

In addition to the commands used to configure the router for GGSN support, the GGSN feature supports several optional commands that you can use to customize your GTP configuration.

For certain GTP processing options, the default values represent recommended values. Other optional commands also are set to default values, but Cisco Systems recommends modifying these commands to optimize your network as necessary, or according to your router hardware. This section describes some of the commands that you should consider optimizing for GTP signaling.

To optimize your GTP signaling configuration, use the following commands beginning in global configuration mode:

Command	Purpose
`Router(config)# gprs gtp n3-requests requests`	(Optional) Specifies the maximum number of times that the GGSN attempts to send a signaling request. The default is 5.
`Router(config)# gprs gtp path-echo-interval interval`	(Optional) Specifies the number of seconds that the GGSN waits before sending an echo-request message to check for GTP path failure. The default is 60 seconds.
`Router(config)# gprs gtp t3-response response_interval`	(Optional) Specifies the the initial number of seconds that the GGSN waits before resending a signaling request message when a response to a request has not been received. This time is doubled for every retry. The default is 1 second.

Note These GTP signaling commands are also used to support echo timing on the GGSN. For more information about echo timing on the GGSN, see the "Configuring Echo Timing on the GGSN" section on page 4-3.

Configuring Other GTP Signaling Options

This section describes some of the other GTP signaling options that you can modify as necessary to support your network needs.

To configure some of the other GTP signaling options, use the following commands beginning in global configuration mode:

Command	Purpose
`Router(config)# gprs gtp map signalling tos tos-value`	(Optional) Specifies an IP ToS mapping for GTP signaling packets. The default is 5.
`Router(config)# gprs gtp n3-buffer-size bytes`	(Optional) Specifies the size of the receive buffer that the GGSN uses to receive GTP signaling messages and packets sent through the tunneling protocol. The default is 8192 bytes.

Configuring the Maximum Number of PDP Contexts on the GGSN

The practical upper limit for the maximum number of PDP contexts supported on the GGSN varies by router platform, amount of memory installed, and the type of configuration configured (whether a method of Point to Point Protocol [PPP] has been configured to forward packets beyond the terminal equipment and mobile termination and the rate of PDP context creation to be supported). The default maximum is 10000.

The following list shows the maximum number of PDP contexts that are supported on the GGSN according to the memory and router series in use when a method of PPP has not been configured:

- Cisco 7206 VXR NPE-300 router with 256 Mb RAM—80,000 PDP contexts.
- Cisco 7206 VXR NPE-400 router with 512 Mb RAM—135,000 PDP contexts.

For information on the maximum number of PDP contexts supported on the GGSN when a method of PPP has been configured, see "Overview of PPP Support on the GGSN" section on page 7-1.

Note When the maximum allowable number of PDP contexts is reached, the GGSN refuses new PDP contexts (mobile sessions) until sessions are available.

To configure the maximum number of PDP contexts on the GGSN, use the following command beginning in global configuration mode:

Command	Purpose
`Router(config)# gprs maximum-pdp-context-allowed pdp-contexts`	Specifies the maximum number of PDP contexts (mobile sessions) that can be activated on the GGSN.

Configuring the Maximum Number of PDP Contexts When Using DFP with Load Balancing

If you use Dynamic Feedback Protocol (DFP) with GPRS/UMTS load balancing, you must also specify a maximum number of PDP contexts for each GGSN. Do not accept the default value of 10000 PDP contexts; a value of 45000 is recommended. Significantly lower values can impact performance in a GPRS/UMTS load-balancing environment.

Note For more information about configuring GPRS/UMTS load balancing, see the *IOS Server Load Balancing*, 12.1(9)E documentation located at Cisco.com at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121limit/121e/121e9/index.htm

To configure the maximum number of PDP contexts on the GGSN for DFP, use the following command beginning in global configuration mode:

Command	Purpose
`Router(config)# gprs maximum-pdp-context-allowed 45000`	Specifies 45000 as the maximum number of PDP contexts (mobile sessions) that can be activated on the GGSN.

Controlling Idle Sessions on the GGSN

GPRS/UMTS provides always-on services for mobile users. Sessions can be established with the GGSN that provide network connectivity, even though no activity may be occurring over that session. Once a PDP context is established on the GGSN, whether there is activity over the session or not, resources are being used by the GGSN. Therefore, you might want to control the amount of time that a mobile session can remain inactive on the GGSN before the PDP context is cleared. The GGSN can only support a certain number of PDP contexts. The number of PDP contexts supported depends upon the configuration and memory resources of the router.

This section describes how you can configure the idle timer on the GGSN to control when the GGSN deletes idle PDP contexts. It includes the following topics:

Overview of the Idle Timer on the GGSN

The GGSN allows you to control the clearing of inactive PDP contexts by configuring an idle timer. The idle timer specifies the amount of time that the GGSN waits before purging idle mobile sessions. When the session reaches the timeout value, the PDP context is deleted. By default, the GGSN clears any idle session after 72 hours.

You can configure the idle timer globally on the GGSN for sessions occurring on all access points, and you can configure an idle timer for a particular access point. In addition to the idle timer that you can configure on the GGSN, RADIUS servers can also specify session timeout attributes for a PDP context.

The following list describes the order in which the GGSN implements the idle timer:

1. RADIUS server—If the access point is configured for non-transparent access mode and the RADIUS server returns a session timeout attribute, then the GGSN uses the session idle timeout value from the RADIUS server.
2. Access-point—If the access point is configured for transparent access mode, or is in non-transparent access mode and the RADIUS server does not return a session idle timeout value, then the GGSN uses the value that you specified for the **session idle-time** command.
3. Global timer—If the GGSN does not receive a session idle timeout value from the RADIUS server or the access point, then it uses the value that you specified in the **gprs idle-pdp-context purge-timer** command.

In summary, the idle timeout value from the RADIUS server takes precedence over the idle timer configuration on the GGSN, and the idle timer for a particular access point takes precedence over the globally configured idle timer.

The **session idle-time** command value overrides the value configured in the **gprs idle-pdp-context purge-timer** command for that access-point.

Configuring the Idle Timer Globally on the GGSN

To configure the amount of time that the GGSN waits before purging idle sessions on the GGSN for all access points, use the following command beginning in global configuration mode:

Command	Purpose
`Router(config)# gprs idle-pdp-context purge-timer hours`	Specifies the time (between 0 and 255 hours) that the GGSN waits before purging idle mobile sessions for all access points. The default value is 72 hours.

Configuring the Idle Timer for an Access Point on the GGSN

To configure the amount of time that the GGSN waits before purging idle sessions on the GGSN for a particular access point, use the following command beginning in access-point configuration mode:

Command	Purpose
`Router(config-access-point)# session idle-time hours`	Specifies the time (between 1 and 168 hours) that the GGSN waits before purging idle mobile sessions at the access point. The idle timer at an access point is disabled by default.

Note When you enable the session idle timer, any G-CDRs triggered for the termination of a PDP context because of the expiration of the purge timer will have a cause value of "normal."

Disabling the Idle Timer on the GGSN

By default, the GGSN purges idle mobile session after 72 hours for all access points. If you want to allow mobile sessions to remain inactive for an indefinite period of time, or if you want to specify the idle timer individually at each access point, you can disable the global idle timer by specifying a value of 0 in the **gprs idle-pdp-context purge-timer** command. By default, the access-point level idle timer is disabled by default with a value of 0.

To disable the idle timer on the GGSN for all access points, use the following command beginning in global configuration mode:

Command	Purpose
`Router(config)# gprs idle-pdp-context purge-timer 0`	Specifies 0 hours for the idle timer, which means that the GGSN does not delete idle PDP contexts.

Verifying the Idle Timer Configuration

To display idle timer information for a particular PDP context, you can use the **show gprs gtp pdp-context** command using the **tid** or **imsi** keywords. The following example shows sample output for the **show gprs gtp pdp-context tid** command for a PDP context with an idle timer value of 200 hours. The idle timer value is displayed in the gtp pdp idle time field shown in bold:

```
Router# show gprs gtp pdp-context tid 1234567812345678
TID              MS Addr         Source  SGSN Addr       APN
1234567812345678 10.106.0.119    Radius  10.40.40.2      www.pdn.com

    current time :Jan 04 2002 02:18:12
    user_name (IMSI): 214365872143658    MS address: 10.106.0.119
    MS International PSTN/ISDN Number (MSISDN): 9987876565
    sgsn_addr_signal: 10.40.40.2         ggsn_addr_signal: 10.29.29.1
    signal_sequence:  1                  seq_tpdu_up:      0
    seq_tpdu_down:    0
    upstream_signal_flow:   39           upstream_data_flow:   40
    downstream_signal_flow: 1            downstream_data_flow: 1
    RAupdate_flow:          0
    pdp_create_time:   Jan 04 2002 02:18:02
    last_access_time:  Jan 04 2002 02:18:02
    mnrgflag:          0                 tos mask map: 40
    gtp pdp idle time: 200
    gprs qos_req: 091101               canonical Qos class(req.): 03
    gprs qos_neg: 0A1101               canonical Qos class(neg.): 03
    effective bandwidth: 800
    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:         222452491
    pdp reference count: 2
    ntwk_init_pdp:       0
```

Configuring Flow Control for GTP Error Messages

By default, the GGSN disables flow control for GTP error messages. You can enable flow control for transmission of GTP error messages using the **gprs gtp error-indication-throttle** global configuration command. This command sets the initial value of a counter which is decremented each time an error indication message is sent. When the counter reaches zero, the GGSN stops transmitting error indication messages. The GGSN resets this counter to the configured throttle value after one second.

To configure flow control for GTP error messages, use the following command beginning in global configuration mode:

Command	Purpose
`Router(config)# gprs gtp error-indication-throttle window-size` *size*	(Optional) Specifies the maximum number of error indication messages that the GGSN sends out in one second, where size is an integer between 0 and 256. There is no default value.

Monitoring and Maintaining GTP on the GGSN

This section provides a summary list of the **show** commands that you can use to monitor GTP on the GGSN.

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

Command	Purpose
Router# **show gprs gtp parameters**	Displays information about the current GTP configuration on the GGSN.
Router# **show gprs gtp path** {**remote-address** *ip-address* [*remote-port-num*] | **version** *gtp-version* | **all**}	Displays information about one or more GTP paths between the GGSN and other GPRS/UMTS devices.
Router# **show gprs gtp pdp-context** {**tid** *tunnel_id* | ms-address *ip_address* [**apn-index** *access-point-index*] | **imsi** *imsi* [**nsapi** *nsapi* [**tft**]] | **path** *ip-address* [*remote_port_num*] | **access-point** *access-point-index* | **pdp-type** {**ip** | **ppp**} | **qos-umts-class** {**background** | **conversational** | **interactive** | **streaming**} | **qos** {**precedence** {**low** | **normal** | **high**} | **qos-delay** {**class1** | **class2** | **class3** | **classbesteffort**} | **version** *gtp-version*} | **all**}	Displays a list of the currently active PDP contexts (mobile sessions). **Note** The **show gprs gtp pdp-context** command options vary depending on the type of QoS method that is enabled on the GGSN.
Router# **show gprs gtp ms** {**imsi** *imsi* | **access-point** *access-point-index* | **all**}	Displays a list of the currently active MSs on the GGSN.
Router# **show gprs gtp statistics**	Displays the current GTP statistics for the GGSN (such as IE, GTP signaling, and GTP PDU statistics).
Router# **show gprs gtp status**	Displays information about the current status of GTP on the GGSN.

Configuration Examples

This section includes the following examples:

- GGSN Configuration Example, page 4-19
- Dynamic Echo Timer Configuration Example, page 4-20

GGSN Configuration Example

The following example shows part of a sample GGSN configuration with some of the commands that you use to configure basic GGSN GTP services:

```
Router# show running-config

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
!
! Enables GGSN services
```

```
!
service gprs ggsn
!
. . .
!
!
! Configures a loopback interface
!
interface loopback 1
 ip address 10.40.40.3 255.255.255.0
!
! Defines the virtual-template interface
! with GTP encapsulation
!
interface Virtual-Template1
 ip unnumber loopback 1
 encapsulation gtp
 gprs access-point-list gprs
!
. . .
!
gprs access-point-list gprs
!
  access-point 1
   access-point-name gprs.cisco.com
   exit
!
  access-point 2
   access-point-name gprt.cisco.com
   exit
   !
  access-point 3
   access-point-name gpru.cisco.com
   access-mode non-transparent
   aaa-group authentication foo
   exit
!
! Configures GTP parameters
!
gprs maximum-pdp-context-allowed 90000
gprs gtp path-echo-interval 0
gprs default charging-gateway 10.15.15.1
!
. . .
!
end
```

Dynamic Echo Timer Configuration Example

The following example shows part of a sample GGSN configuration for the dynamic echo timer. In this example, the dynamic echo timer is enabled, the smooth factor is changed from the default of 2 to 5, and the dynamic minimum value is changed from the default of 5 seconds to 10 seconds:

```
Router# show running-config

Current configuration : 6769 bytes
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
```

```
no service password-encryption
service internal
service gprs ggsn
!
. . .
!
interface loopback 1
 ip address 10.41.41.1 255.255.255.0
!
interface Virtual-Template1
 ip unnumber loopback 1
 encapsulation gtp
 gprs access-point-list gprs
!
. . .
!
gprs access-point-list gprs
  access-point 1
   access-point-name gprs.cisco.com
   exit
   !
  access-point 2
   access-point-name gprt.cisco.com
   access-mode non-transparent
   aaa-group authentication test2
   aaa-group accounting test2
   ip-address-pool dhcp-proxy-client
   dhcp-server 10.65.0.1
   dhcp-gateway-address 10.65.0.1
   exit
!
! Enables the dynamic echo timer
!
gprs gtp echo-timer dynamic enable
!
! Configures a smooth factor of 5
!
gprs gtp echo-timer dynamic smooth-factor 5
!
! Configures the dynamic minimum as 10 seconds
!
gprs gtp echo-timer dynamic minimum 10
gprs gtp response-message wait-accounting
!
end
```

Configuring Charging on the GGSN

This chapter describes how to configure the charging function on the GGSN. Charging processing is enabled by default on the GGSN. There are several ways to customize communication with a charging gateway. Many of the default values for the charging options will provide a satisfactory configuration until you become more familiar with your network and decide to customize the charging interface.

For a complete description of the GGSN commands in this chapter, refer to the *Cisco IOS Mobile Wireless 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:

Configuring a Physical Interface to the Charging Gateway

To establish access to an external charging gateway in the GPRS/UMTS network, you must configure a physical interface on the GGSN to connect to the network of the charging gateway. In GPRS/UMTS, the interface between the GGSN and the charging gateway is referred to as the Ga interface.
GGSN Release 4.0 supports both a 2.5G Ga interface and 3G Ga interface.

For more information about configuring physical interfaces on Cisco Systems' routers, see the *Cisco IOS Interface Configuration Guide* and the *Cisco IOS Interface Command Reference*.

To configure a physical interface to the charging gateway that supports Fast Ethernet on a Cisco 7200 series router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type slot/port`	Defines a physical interface on the GGSN, where *type* is **fastethernet**, and *slot/port* is the hardware slot and port on the interface.
Step 2	`Router(config-if)# ip address ip-address mask [secondary]`	Specifies an IP address for the interface, where: • *ip-address*—Specifies the IP address of the interface in dotted decimal format. • *mask*—Specifies a subnet mask in dotted decimal format. • **secondary**—Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 3	`Router(config-if)# ip route-cache cef`	(Optional) Enables CEF operation on an interface.

Verifying Interface Configuration to the Charging Gateway

To verify the physical interface to the charging gateway you can first verify your GGSN configuration and then verify that the interface is available.

Step 1 To verify that you have properly configured a Ga interface on the GGSN, use the **show running-config** command. The following example is a portion of the output from the command showing the FastEthernet5/1 physical interface configuration as the Gn interface to the SGSN:

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

Current configuration : 2875 bytes
!
version 12.2
. . .
!
interface FastEthernet5/1
 description Ga interface
 ip address 10.9.0.1 255.255.255.0
 no ip mroute-cache
 duplex full
. . .
```

Step 2 To verify that a physical interface is available, use the **show ip interface brief** command. The following example shows that the FastEthernet5/1 interface to the charging gateway is in “up” status and the protocol is also “up”:

```
Router #show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            10.10.1.3       YES NVRAM  up                    up
FastEthernet1/0            10.29.0.2       YES NVRAM  up                    up
FastEthernet2/0            unassigned      YES NVRAM  administratively down down
FastEthernet5/1            10.9.0.1        YES NVRAM  up                    up
```

```
Ethernet6/0                10.99.0.12      YES NVRAM  up                    up
Ethernet6/1                unassigned      YES NVRAM  administratively down down
Ethernet6/2                unassigned      YES NVRAM  administratively down down
Ethernet6/3                unassigned      YES NVRAM  administratively down down
Ethernet6/4                unassigned      YES NVRAM  administratively down down
Ethernet6/5                unassigned      YES NVRAM  administratively down down
Ethernet6/6                unassigned      YES NVRAM  administratively down down
Ethernet6/7                10.35.35.2      YES NVRAM  up                    up
Virtual-Access1            10.44.44.1      YES TFTP   up                    up
Virtual-Template1          10.44.44.1      YES manual down                  down
```

Configuring the Charging Gateway

To configure the default charging gateway, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`gprs default charging-gateway`** {*ip-address* | *name*} [{*ip-address* | *name*}]	Specifies a primary charging gateway (and backup), where: • *ip-address*—Specifies the IP address of a charging gateway. The second (optional) *ip-address* argument specifies the IP address of a secondary charging gateway. • *name*—Specifies the host name of a charging gateway. The second (optional) *name* argument specifies the host name of a secondary charging gateway.

Changing the Default Charging Gateway

To change the default charging gateway, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`gprs default charging-gateway`** `10.9.0.2`	Specifies a primary charging gateway at IP address 10.9.0.2.
Step 2	`Router(config)#` **`no gprs default charging-gateway`** `10.9.0.2`	Removes the primary charging gateway at IP address 10.9.0.2.
Step 3	`Router(config)#` **`gprs default charging-gateway`** `10.9.0.3`	Specifies the new default primary charging gateway at IP address 10.9.0.3.

Configuring the Transport Protocol for the Charging Gateway

You can configure the GGSN to support either Transport Control Protocol (TCP) or User Datagram Protocol (UDP) as the transport path protocol for communication with the charging gateway.

The GGSN default configuration specifies UDP, which is a connectionless protocol that is considered an unreliable transport method but can yield greater performance.

Configuring TCP as the Charging Gateway Path Protocol

TCP is a connection-based protocol that provides reliable transmission through packet acknowledgment. To specify TCP as the transport path protocol, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gprs charging cg-path-requests 1`	Specifies the number of minutes that the GGSN waits before trying to establish the TCP path to the charging gateway when TCP is the specified path protocol. The default is 0 minutes, which disables the timer.
Step 2	`Router(config)# gprs charging path-protocol tcp`	Specifies that the TCP networking protocol is used by the GGSN to transmit and receive charging data.

Configuring UDP as the Charging Gateway Path Protocol

The GGSN default configuration specifies UDP as the transport path protocol to the charging gateway. If you need to reconfigure the charging gateway for UDP transport, use the following command in global configuration mode:

Command	Purpose
`Router(config)# gprs charging path-protocol udp`	Specifies that the UDP networking protocol is used by the GGSN to transmit and receive charging data. The default value is UDP.

Configuring the Charging Release

GGSN Release 4.0 and later supports both 2.5G and 3G Ga interfaces and GPRS (R97/R98) and UMTS (R99) QoS profile formats.

Depending on the CG and GGSN configuration, the following actions take place:

- If the GGSN is configured to present R97/R98 G-CDRs and the PDP context is R99, the GGSN will present a R97/R98 G-CDR by converting the R99 QoS profile to an R97/R98 QoS profile.
- If the GGSN is configured to present R99 G-CDRs and the PDP context is R98, the GGSN will present a R99 G-CDR by converting the QoS profile.

- If the GGSN is configured to present R97/R98 G-CDRS and the PDP context is R98, the GGSN presents an R97/R98 G-CDR.
- If the GGSN is configured to present R99 G-CDRs and the PDP context is R99, the GGSN presents an R99 G-CDR.

To configure the G-CDR version presented by the GGSN, use the following command in global configuration mode:

Command	Purpose
`Router(config)# gprs charging release {99 | 98}`	Specifies that the GGSN present R97/R98 and R99 QoS profile formats in G-CDRs or presents only R97/R98 QoS profile formats. The default value is 99.

Configuring Charging for Roamers

The GGSN can be configured to generate G-CDRs for roaming mobile subscribers.

When the charging for roamers feature is enabled on the GGSN and a create PDP context request is received, the Information Element (IE) containing the SGSN Signalling Address field is matched against the list of PLMN IP address ranges that have been defined using the **gprs plmn ip address** command. If the GGSN determines that the SGSN that sent the create PDP context request is not located within the same PLMN as it is, it generates a CDR. If the GGSN does determine that the SGSN is located in the same PLMN, it will not generate a CDR until it receives notification that the SGSN has changed to that of one located in another PLMN.

To enable charging for roamers on the GGSN using the **gprs charging roamers command**, you should first define a set of IP address ranges for a PLMN using the **gprs plmn ip address** command.

Note It is important that you configure the **gprs plmn ip address** and **gprs charging roamers** commands in their proper order. After you configure the IP address range for a PLMN, use the **gprs charging roamers** command to enable the charging for roamers feature on the GGSN. You can change the IP address range by reissuing the **gprs plmn ip address** command.

To verify your configuration, use the **show gprs charging parameters** command to see if the charging for roamers command is enabled. To verify your PLMN IP address ranges, use the **show gprs plmn ip address** command.

Configuring PLMN IP Address Ranges

Depending on how PLMN IP address ranges have been defined using the **gprs plmn ip address** *start_ip end_ip* [**sgsn**] command, the charging for roamers feature operates as follows:

- If no PLMN IP address ranges have been configured using the **gprs plmn ip address** *start_ip end_ip* [**sgsn**] command, the GGSN generates CDRs for all initiated PDP contexts regardless of whether the GGSN and SGSN are located within the same PLMN.
- If a list of PLMN IP address ranges has been configured using the **gprs plmn ip address** *start_ip end_ip* [**sgsn**] command, but the **sgsn** keyword has not been specified for any of the ranges, the GGSN uses all the range entries to determine whether the SGSN is located within the same PLMN.

- If a list of PLMN IP address ranges has been configured using the **gprs plmn ip address** *start_ip end_ip* [**sgsn**] command, and one or more of those ranges has been defined using the **sgsn** key word, the GGSN uses those ranges defined with the **sgsn** keyword to determine whether an SGSN is located within the same PLMN.

  With this configuration, the following scenarios outline how the charging for roamers feature will function:

  – MS1 is subscribed to PLMN1 and attaches to an SGSN in PLMN2. From PLMN2, MS1 initiates a PDP context with the GGSN in PLMN1. In this case, MS1 is a roamer and the GGSN generates a CDR because it determines that the SGSN is located in a different PLMN.

  – MS1 is subscribed to PLMN1 and attaches to an SGSN in PLMN2. From PLMN2, MS1 initiates a PDP context with the GGSN in PLMN2. In this case, MS1 is not a roamer because the SGSN and GGSN are in the same PLMN. The GGSN does not create a G-CDR.

To configure PLMN IP address ranges, use the following command in global configuration mode.

Command	Purpose
`Router(config)# gprs plmn ip address start_ip end_ip [sgsn]`	Specifies the IP address range of a PLMN.

Enabling Charging for Roamers

To enable the charging for roamers feature on the GGSN, use the following command in global configuration mode:

Command	Purpose
`Router(config)# gprs charging roamers`	Enables charging for roamers on the GGSN.

Customizing the Charging Gateway

For the GGSN charging options, the default values represent recommended values. Other optional commands also are set to default values, but Cisco Systems recommends modifying these commands to optimize your network as necessary, or according to your router hardware.

The GGSN uses echo timing to maintain the path between SGSNs and external charging gateways. However, the GGSN can only implement a single method of echo timing for all of the paths it needs to maintain. To learn more about echo timing on the GGSN, or to modify the echo timing feature, see the "Configuring Echo Timing on the GGSN" section on page 4-3 in the "Configuring GGSN GTP Services" chapter.

Use the following global configuration commands to fine-tune charging processing on the GGSN:

Command	Purpose
`Router(config)# gprs charging cdr-aggregation-limit CDR_limit`	Specifies the maximum number of CDRs that the GGSN aggregates in a charging data transfer message to a charging gateway. The default is 255 CDRs.
`Router(config)# gprs charging cdr-option apn-selection-mode`	Enables the GGSN to provide the reason code for APN selection in G-CDRs. This is disabled by default.

Command	Purpose
Router(config)# **gprs charging cdr-option local-record-sequence-number**	Enables the GGSN to use the local record sequence number field in G-CDRs. This is disabled by default.
Router(config)# **gprs charging cdr-option node-id**	Enables the GGSN to specify the node that generated the CDR in the node ID field in G-CDRs. This is disabled by default.
Router(config)# **gprs charging cdr-option no-partial-cdr-generation**	Disables the GGSN from creating non-primary partial G-CDRs. The default is non-primary partial CDR creation is enabled. **Note** Enable this feature only when there are no active PDP contexts. Enabling this feature will affect all subsequent PDP contexts.
Router(config)# **gprs charging cdr-option packet-count**	Enables the GGSN to provide uplink and downlink packet counts in the optional record extension field in G-CDRs. This is disabled by default.
Router(config)# **gprs charging cdr-option served-msisdn**	Enables the GGSN to provide the MSISDN number from the create PDP context request in G-CDRs. This is disabled by default.
Router(config)# **gprs charging cg-path-requests** *minutes*	Specifies the number of minutes that the GGSN waits before trying to establish the TCP path to the charging gateway when TCP is the specified path protocol. The default is 0 minutes, which disables the timer.
Router(config)# **gprs charging container change-limit** *number*	Specifies the maximum number of charging containers within each G-CDR from the GGSN. The default is 5.
Router(config)# **gprs charging container sgsn-change-limit** *number*	Specifies the maximum number of SGSN changes that can occur before closing a G-CDR for a particular PDP context. The default is disabled.
Router(config)# **gprs charging container volume-threshold** *threshold_value*	Specifies the maximum number of bytes that the GGSN maintains in a user's charging container before closing it and updating the G-CDR. The default is 1,048,576 bytes (1 MB).
Router(config)# **gprs charging disable**	Disables charging transactions on the GGSN. Charging is enabled by default.
Router(config)# **gprs charging flow-control private-echo**	Implements an echo request with private extensions for maintaining flow control on packets transmitted to the charging gateway. This is disabled by default.
Router(config)# **gprs charging header short**	Enables the GGSN to use the GTP short header (6-byte header) instead of the GTP long header. This is disabled by default.
Router(config)# **gprs charging map data tos** *tos_value*	Specifies an IP ToS mapping for GPRS charging packets. The default is 3.
Router(config)# **gprs charging packet-queue-size** *queue_size*	Specifies the maximum number of unacknowledged charging data transfer requests that the GGSN maintains in its queue. The default is 128 packets.
Router(config)# **gprs charging path-protocol** {**udp** | **tcp**}	Specifies the protocol that the GGSN uses to transmit and receive charging data. The default is UDP.
Router(config)# **gprs charging port** *port-num*	Configures the destination port of the charging gateway. The default is 3386.
Router(config)# **gprs charging send-buffer** *bytes*	Configures the size of the buffer that contains the GTP' PDU and signaling messages on the GGSN. The default is 1460 bytes.

Command	Purpose
`Router(config)# gprs charging server-switch-timer seconds`	Specifies a timeout value that determines when the GGSN attempts to find an alternate charging gateway after a destination charging gateway cannot be located or becomes unusable. The default is 60 seconds.
`Router(config)# gprs charging tariff-time time`	Specifies a time of day when GPRS/UMTS charging tariffs change. There is no default tariff time.
`Router(config)# gprs charging message transfer-request command-ie`	Specifies for the GGSN to include the Packet Transfer Command IE in Data Record Transfer Response messages. **Note** GGSN 4.0 supports the Send Data Record Packet command.
`Router(config)# gprs charging message transfer-response number-responded`	Specifies for the GGSN to use the Number of Requests Responded field instead of the Length field in the Requests Responded IE of Data Record Transfer Response messages. This is disabled by default.
`Router(config)# gprs charging transfer interval seconds`	Specifies the number of seconds that the GGSN waits before it transfers charging data to the charging gateway. The default is 105 seconds.

For information about configuring GGSN GTP options, see the "Customizing the GGSN Configuration" section on page 4-13 in the "Configuring GGSN GTP Services" chapter.

Disabling Charging Processing

Caution The **gprs charging disable** command removes charging data processing on the GGSN, which means that the data required to bill customers for network usage is not being collected by the GGSN nor sent to the charging gateway. Cisco Systems, Inc. recommends that you avoid using this command in production GPRS/UMTS network environments. When necessary to use this command, use it with extreme care and reserve its usage only under non-production network conditions.

You can disable charging on the GGSN only when all open CDRs have been processed and sent to the charging gateway. To clear the current GGSN CDRs, use the **clear gprs charging cdr** privileged EXEC command.

To disable charging processing on the GGSN, use the following command beginning in global configuration mode:

Command	Purpose
`Router(config)# gprs charging disable`	Disables charging transactions on the GGSN.

Monitoring and Maintaining Charging on the GGSN

This section provides a summary list of the **show** commands that you can use to monitor charging functions on the GGSN.

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

Command	Purpose
Router# **show gprs charging parameters**	Displays information about the current GGSN charging configuration.
Router# **show gprs charging statistics**	Displays cumulative statistics about the transfer of charging packets between the GGSN and charging gateways.
Router# **show gprs charging status** {**tid** *tunnel_id* | **access-point** *access-point-index* | **all**}	Displays current statistics about the transfer of charging packets between the GGSN and charging gateways.

Configuration Example

The following configuration example shows part of a sample GGSN configuration with some of the commands that you use to configure charging services:

```
Router# show running-config
service gprs ggsn
!
. . .
!
interface Ethernet5/1
 description Ga interface
 ip address 10.9.0.1 255.255.0.0
 duplex half
!
. . .
!
interface loopback 1
 ip address 10.40.40.1 255.255.255.0
!
interface Virtual-Template1
 ip unnumber loopback 1
 encapsulation gtp
 gprs access-point-list gprs
!
. . .
!
gprs access-point-list gprs
  access-point 1
   access-point-name auth-accounting
   access-mode non-transparent
   aaa-group authentication first
   aaa-group accounting second
   ip-address-pool dhcp-proxy-client
   dhcp-server 10.60.0.1
   dhcp-gateway-address 10.60.0.1
   exit
   !
. . .
!
gprs default charging-gateway 10.9.0.2
```

```
gprs charging send-buffer 1000
gprs charging container volume-threshold 500000
gprs charging container change-limit 3
gprs charging cdr-aggregation-limit 10
gprs charging cdr-option apn-selection-mode
gprs charging cdr-option served-msisdn
!
. . .
!
end
```

Configuring Network Access to the GGSN

This chapter describes how to configure access from the GGSN to a SGSN, packet data network (PDN), and optionally to a virtual private network (VPN). It also includes information about configuring access points on the GGSN.

For a complete description of the GGSN commands in this chapter, refer to the *Cisco IOS Mobile Wireless 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:

Configuring a Physical Interface to the SGSN

The type of physical interface that you configure on the GGSN depends on whether you are supporting an SGSN that is collocated with a GGSN, or an enterprise GGSN that is connected to the SGSN through a WAN interface.

When a GGSN is collocated with the SGSN, the physical interface is frequently configured for Fast Ethernet. The supported WAN interfaces for a remote SGSN include T1/E1, T3/E3, and Frame Relay.

For more information about configuring physical interfaces on Cisco Systems' routers, see the *Cisco IOS Interface Configuration Guide* and the *Cisco IOS Interface Command Reference*.

To configure a physical interface to the SGSN that supports Fast Ethernet on a Cisco 7200 series router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type slot/port`	Defines a physical interface on the GGSN, where *type* is **fastethernet**, and *slot/port* is the hardware slot and port on the interface.
Step 2	`Router(config-if)# ip address ip-address mask [secondary]`	Specifies an IP address for the interface, where: • *ip-address*—Specifies the IP address of the interface in dotted decimal format. • *mask*—Specifies a subnet mask in dotted decimal format. • **secondary**—Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 3	`Router(config-if)# ip route-cache cef`	(Optional) Enables CEF operation on an interface.

Verifying Interface Configuration to the SGSN

To verify the physical interface to the SGSN you can first verify your GGSN configuration and then verify that the interface is available.

Step 1 To verify that you have properly configured a Gn interface on the GGSN, use the **show running-config** command. The following example is a portion of the output from the command showing the FastEthernet0/0 physical interface configuration as the Gn interface to the SGSN:

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

Current configuration : 2875 bytes
!
version 12.2
. . .
!
interface FastEthernet0/0
 description Gn interface to SGSN
 ip address 10.10.1.3 255.255.255.0
 no ip mroute-cache
 duplex full
. . .
```

Step 2 To verify that a physical interface is available, use the **show ip interface brief** command. The following example shows that the FastEthernet0/0 interface to the SGSN is in “up” status and the protocol is also “up”:

```
Router #show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            10.10.1.3       YES NVRAM  up                    up
FastEthernet1/0            10.29.0.2       YES NVRAM  up                    up
FastEthernet1/1            10.13.0.2       YES NVRAM  up                    up
FastEthernet2/0            unassigned      YES NVRAM  administratively down down
```

```
Ethernet6/0                10.99.0.12      YES NVRAM  up                    up
Ethernet6/1                unassigned      YES NVRAM  administratively down down
Ethernet6/2                unassigned      YES NVRAM  administratively down down
Ethernet6/3                unassigned      YES NVRAM  administratively down down
Ethernet6/4                unassigned      YES NVRAM  administratively down down
Ethernet6/5                unassigned      YES NVRAM  administratively down down
Ethernet6/6                unassigned      YES NVRAM  administratively down down
Ethernet6/7                10.35.35.2      YES NVRAM  up                    up
Virtual-Access1            10.44.44.1      YES TFTP   up                    up
Virtual-Template1          10.44.44.1      YES manual down                  down
```

Configuring a Route to the SGSN

To communicate with the SGSN, you can use static routes or a routing protocol, such as Open Shortest Path First (OSPF).

Note For the SGSN to communicate successfully with the GGSN, the SGSN must also configure a static route, or be able to dynamically route to the IP address of the GGSN *virtual template,* not the IP address of a GGSN physical interface.

The following sections provide some basic commands that you can use to configure a static route or enable OSPF routing on the GGSN. For more information about configuring IP routes, see the *Cisco IOS IP Configuration Guide* and *Cisco IOS IP Command References*.

The following topics are included in this section:

Configuring a Static Route to the SGSN

A static route establishes a fixed route between the GGSN and the SGSN that is stored in the routing table. If you are not implementing a routing protocol, such as OSPF, then you can configure a static route from the GGSN to the SGSN, to establish the path between these network devices.

To configure a static route from a physical interface on the GGSN to the SGSN, use the following commands beginning in global configuration mode:

Command	Purpose
`Router(config)# ip route prefix mask {ip-address | interface-type interface-number} [distance] [tag tag] [permanent]`	Configures a static IP route, where: • *prefix*—Specifies the IP route prefix for the destination. (This is the IP address of the SGSN.) • *mask*—Specifies the prefix mask for the destination. (This is the subnet mask of the SGSN network.) • *ip-address*—Specifies the IP address of the next hop that can be used to reach the destination network. • *interface-type interface-number*—Specifies the network interface type and interface number that can be used to reach the destination network. (This is a physical interface on the GGSN for the Gn interface.) • *distance*—Specifies an administrative distance for the route. • **tag** *tag*—Specifies a tag value that can be used as a "match" value for controlling redistribution via route maps. • **permanent**—Specifies that the route will not be removed, even if the interface shuts down.

Configuring OSPF on the GGSN

As with other routing protocols, enabling OSPF requires that you create an OSPF routing process, specify the range of IP addresses to be associated with the routing process, and assign area IDs to be associated with that range of IP addresses.

To configure OSPF, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router ospf process-id`	Enables OSPF routing, and enters router configuration mode, where *process-id* specifies an internally used identification parameter for an OSPF routing process. The *process-id* is locally assigned and can be any positive integer. A unique value is assigned for each OSPF routing process.
Step 2	`Router(config-router)# network ip-address wildcard-mask area area-id`	Defines an interface on which OSPF runs and defines the area ID for that interface, where: • *ip-address*—Specifies the IP address to be associated with the OSPF network area. • *wildcard-mask*—Specifies the IP address mask that includes “don't care” bits for the OSPF network area. • *area-id*—Specifies the area that is to be associated with the OSPF address range. It can be specified as either a decimal value or as an IP address. If you intend to associate areas with IP subnets, you can specify a subnet address as the area-id.

Verifying the Route to the SGSN

To verify the route to the SGSN you can first verify your GGSN configuration and then verify that a route has been established.

Step 1 To verify the GGSN configuration, use the **show running-config** command and verify the static route that you configured to the SGSN, or your OSPF configuration. The following example shows a partial configuration of an OSPF configuration for the 10.10.0.0 network using the FastEthernet0/0 interface to the SGSN:

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

Current configuration : 2875 bytes
!
version 12.2
. . .
!
interface FastEthernet0/0
 description Gn interface to SGSN
 ip address 10.10.1.3 255.255.255.0
 no ip mroute-cache
 duplex full
!
```

```
interface FastEthernet6/0
 ip address 172.16.43.243 255.255.255.240
 no ip mroute-cache
 duplex half
!
!
interface loopback 1
 ip address 10.11.11.1 255.255.255.0
!
interface Virtual-Template1
 ip unnumber loopback 1
 encapsulation gtp
!
router ospf 1
 log-adjacency-changes
 network 10.10.0.0 0.0.255.255 area 0
!
ip default-gateway 172.16.43.241
ip classless
ip route 10.22.22.1 255.255.255.255 FastEthernet2/0
ip route 192.64.0.0 255.0.0.0 172.16.43.241
ip route 172.16.0.0 255.255.0.0 172.16.43.241
no ip http server
no ip pim bidir-enable
. . .
```

Step 2 To verify that the GGSN has established a route to the SGSN, you can use the **show ip route** command as shown in bold in the following example:

```
Router# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.11.11.0/24 is subnetted, 1 subnets
C       10.11.11.0 is directly connected, Virtual-Access1
     172.16.0.0/16 is variably subnetted, 1 subnets, 2 masks
S       172.16.0.0/16 [1/0] via 172.16.43.241
C       172.16.43.243/28 is directly connected, FastEthernet6/0
     10.0.0.0/24 is subnetted, 1 subnets
O       10.10.1.0 [110/2] via 10.10.1.3, 00:00:10, FastEthernet0/0
C       10.10.1.0 is directly connected, FastEthernet0/0
```

Configuring Access Points on the GGSN

Successful configuration of access points on the GGSN requires careful consideration and planning to establish the appropriate access for mobile sessions to external PDNs and private networks.

The following topics are included in this section:

Configuration of access points on the GGSN also requires properly establishing communication with any supporting DHCP and RADIUS servers that you might be using to provide dynamic IP addressing and user authentication functions at the access point.

Details about configuring other services such as DHCP and RADIUS for an access point are discussed in the "Configuring DHCP on the GGSN" and "Configuring Security on the GGSN" chapters.

Overview of Access Points

This section includes the following topics:

Description of Access Points in a GPRS/UMTS Network

The GPRS and UMTS standards define a network identity called an access point name (APN). An APN identifies the part of the network where a user session is established, and in the GPRS/UMTS backbone, it serves as a reference to a GGSN. An APN is configured on and accessible from a GGSN in a GPRS/UMTS network.

An APN can provide access to a public data network (PDN), or a private or corporate network. An APN also can be associated with certain types of services such as Internet access or a Wireless Application Protocol (WAP) service.

The APN is provided by either the mobile station (MS) or by the SGSN to the GGSN in a create PDP context request message when a user requests a session to be established.

To identify an APN, a logical name is defined that consists of two parts:

- Network ID—A mandatory part of the APN that identifies the external network to which a GGSN is connected. The network ID can be a maximum of 63 bytes and must contain at least one label. A network ID of more than one label is interpreted as an Internet domain name. An example of a network ID might be "corporate.com."
- Operator ID—An optional part of the APN that identifies the PLMN in which a GGSN is located. The operator ID contains three decimal-separated labels, where the last label must be "gprs." An example of an operator ID might be "mnc10.mcc200.gprs."

  When the operator ID exists, it is placed after the network id, and corresponds to the DNS name of a GGSN. The maximum length of an APN is 100 bytes. When the operator ID does not exist, a default operator ID is derived from the mobile network code (MNC) and mobile country code (MCC) information contained in the international mobile subscriber identity (IMSI).

Access Point Implementation on the Cisco Systems GGSN

Configuring access points is one of the central configuration tasks on the Cisco Systems GGSN. Proper configuration of access points is essential to successful implementation of the GGSN in the GPRS/UMTS network.

To configure APNs, the Cisco GGSN software uses the following configuration elements:

- Access point list—Logical interface that is associated with the virtual template of the Cisco Systems GGSN. The access point list contains one or more access points.
- Access point—Defines an APN and its associated access characteristics, including security and method of dynamic addressing. An access point on the Cisco Systems GGSN can be a virtual or real access point.
- Access point index number—Integer assigned to an APN that identifies the APN within the GGSN configuration. Several of the GGSN configuration commands use the index number to reference an APN.
- Access group—An additional level of security on the router that is configured at an access point to control access to and from a PDN. When an MS is permitted access to the GGSN as defined by a traditional IP access list, the IP access group further defines whether access is permitted to the PDN (at the access point). The IP access group configuration can also define whether access from a PDN to an MS is permitted.

Access Point Types on the GGSN

Cisco IOS GGSN Release 3.0 and later supports the following access point types:

- Real—Use real access point types to configure the GGSN for direct access to a particular target network through a physical interface. The GGSN always uses real access points to reach an external network.
- Virtual—Use virtual access point types to consolidate access to multiple target networks through a virtual APN access point at the GGSN. The GGSN always uses real access points to reach an external network, so virtual access points should be used in combination with real access points on the GGSN.

Cisco IOS GGSN Release 1.4 and earlier only supports real access points.

GGSN Release 3.0 and later supports virtual access point types to address provisioning issues in the PLMN. For more information about configuring virtual access point access to the GGSN from the PLMN, see the "Configuring Virtual APN Access on the GGSN" section on page 6-29.

Basic Access Point Configuration Task List

This section describes the basic tasks that are required to configure an access point on the GGSN. Detailed information about configuring access points for specialized functions such as network-initiated PDP context support, or for virtual APN access are described in separate sections of this chapter.

To configure an access point on the GGSN, perform the following basic tasks:

- Configuring the GPRS Access Point List on the GGSN, page 6-9 (Required)
- Creating an Access Point and Specifying its Type on the GGSN, page 6-9 (Required)
- Configuring Real Access Points on the GGSN, page 6-10 (Required)
  - PDN Access Configuration Task List, page 6-10
  - VPN Access Using VRF Configuration Task List, page 6-12
- Configuring Other Access Point Options, page 6-19 (Optional)

Configuring the GPRS Access Point List on the GGSN

The GGSN software requires that you configure an entity called an access point list. You configure the GPRS access point list to define a collection of virtual and real access points on the GGSN.

When you configure the access point list in global configuration mode, the GGSN software automatically associates the access point list with the virtual template interface of the GGSN. Therefore, the GGSN supports only a single access point list.

Note Be careful to observe that the GPRS access point list and an IP access list are different entities in the Cisco IOS software. A GPRS access point list defines access points and their associated characteristics, and an IP access list controls the allowable access on the router by IP address. You can define permissions to an access point by configuring both an IP access list in global configuration, and configuring the **ip-access-group** command in your access point configuration.

To configure the GPRS access point list and configure access points within it, use the following commands beginning in global configuration mode:

Command	Purpose
`Router(config)# gprs access-point-list list-name`	Specifies a name for a new access point list, or references the name of the existing access point list and enters access-point list configuration mode.

Creating an Access Point and Specifying its Type on the GGSN

You need to define access points within an access point list on the GGSN. Therefore, before you can create an access point, you must define a new access point list, or specify the existing access point list on the GGSN to enter access-point list configuration mode.

When you create an access point you must assign an index number to the access point, specify the domain name (network ID) of the access point, and specify the type of access point (virtual or real). Other options that you can configure for an access point are summarized in the "Configuring Other Access Point Options" section on page 6-19.

To create an access point and specify its type, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gprs access-point-list list-name`	Specifies a name for a new access point list, or references the name of the existing access point list and enters access-point list configuration mode.
Step 2	`Router(config-ap-list)# access-point access-point-index`	Specifies an index number for a new access point definition, or references an existing access point definition, and enters access point configuration mode.

	Command	Purpose
Step 3	`Router(config-access-point)# access-point-name apn-name`	Specifies the network (or domain) name for a PDN that users can access from the GGSN at a defined access point. **Note** The *apn-name* must match the APN that has been provisioned at the MS, HLR, and DNS server.
Step 4	`Router (config-access-point)# access-type {virtual | real}`	(Optional) Specifies the type of access point. The available options are: • **virtual**—APN type that is not associated with any specific physical target network on the GGSN. • **real**—APN type that corresponds to a physical interface to an external network on the GGSN. This is the default value.

Configuring Real Access Points on the GGSN

The GGSN uses real access points to communicate to PDNs or private networks that are available over a Gi interface on the GGSN. Use real access point types to configure the GGSN for direct access to a particular target network through a physical interface.

If you have configured a virtual access point, you must also configure real access points to reach the target networks.

The GGSN supports configuration of access points to public data networks and to private networks. The following sections describe how to configure different types of real access points:

- PDN Access Configuration Task List, page 6-10
- VPN Access Using VRF Configuration Task List, page 6-12

PDN Access Configuration Task List

Configuring a connection to a public packet data network includes the following tasks:

- Configuring an Interface to a PDN (Gi interface) (Required)
- Configuring an Access Point for a PDN (Required)

Configuring an Interface to a PDN

To configure a physical interface to the PDN using Fast Ethernet over the Gi interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type slot/port`	Defines a physical interface on the GGSN, where *type* is **fastethernet**, and *slot/port* is the hardware slot and port on the interface.
Step 2	`Router(config-if)# ip address ip-address mask [secondary]`	Specifies an IP address for the interface, where: • *ip-address*—Specifies the IP address of the interface in dotted decimal format. • *mask*—Specifies a subnet mask in dotted decimal format. • **secondary**—Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 3	`Router(config-if)# ip route-cache cef`	(Optional) Enables CEF operation on an interface. **Note** If you are using VRF for VPN access, you must enable CEF switching on the GGSN. If you enable CEF switching at the global configuration level, then it is automatically enabled for each interface unless it has been specifically disabled at the interface.

Configuring an Access Point for a PDN

To configure an access point for a PDN, you must define a real access point in the GPRS access point list.

To configure a real access point on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gprs access-point-list list-name`	Specifies a name for a new access-point list, or references the name of an existing access-point list and enters access-point list configuration mode.
Step 2	`Router(config-ap-list)# access-point access-point-index`	Specifies an index number for a new access point definition, or references an existing access point definition, and enters access point configuration mode.

	Command	Purpose
Step 3	`Router(config-access-point)# access-point-name apn-name`	Specifies the network (or domain) name for a PDN that users can access from the GGSN at a defined access point. **Note** The *apn-name* must match the APN that has been provisioned at the MS, HLR, and DNS server.
Step 4	`Router(config-access-point)# access-type real`	Specifies an APN type that corresponds to a physical interface to an external network on the GGSN. This is the default value.

For an example of a GPRS access point configuration, see the "Access Point List Configuration Example" section on page 6-49.

VPN Access Using VRF Configuration Task List

The Cisco IOS GGSN software supports connectivity to a virtual private network (VPN) using virtual routing and forwarding (VRF).

The GGSN software provides a couple of ways that you can configure access to a VPN, depending on your network configuration over the Gi interface between the GGSN and your PDNs, and the VPN that you want to access.

To configure VPN access using VRF on the GGSN, perform the following tasks:

- Enabling CEF Switching, page 6-13 (Required)
- Configuring a VRF Routing Table on the GGSN, page 6-13 (Required)
- Configuring a Route to the VPN Using VRF, page 6-13 (Required)
- Configuring an Interface to a PDN Using VRF, page 6-15 (Required)
- Configuring Access to a VPN, page 6-16 (Required)

For a sample configuration, see the "VRF Tunnel Configuration Example" section on page 6-50.

Enabling CEF Switching

When you enable CEF switching globally on the GGSN, all interfaces on the GGSN are automatically enabled for CEF switching. You can also enable CEF switching at a particular interface on the GGSN using the **ip route-cache cef** interface configuration command. For more information about configuring CEF switching, see the "Optimizing GGSN Performance" chapter.

Note To ensure CEF switching functions properly, wait a short period of time before enabling CEF switching after it has been disabled using the **no ip cef** command.

To enable CEF switching for all interfaces on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip cef`	Enables CEF on the route processor card.
Step 2	`Router(config)# gprs gtp ip udp ignore checksum`	Disables verification of the UDP checksum to support CEF switching on the GGSN.

Configuring a VRF Routing Table on the GGSN

To configure a VRF routing table on the GGSN, use the following command beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip vrf vrf-name`	Configures a VRF routing table, and enters VRF configuration mode.
Step 2	`Router(config-vrf)# rd route-distinguisher`	Creates routing and forwarding tables for a VRF and specifies the default route distinguisher for a VPN.

Configuring a Route to the VPN Using VRF

Be sure that a route exists between the GGSN and the private network that you want to access. You can verify connectivity by using the **ping** command from the GGSN to the private network address. To configure a route, you can use a static route or a routing protocol.

Configuring a Static Route Using VRF

To configure a static route using VRF, use the following command beginning in global configuration mode:

Command	Purpose
`Router(config)# ip route vrf vrf-name prefix mask [next-hop-address] [interface {interface-number}] [global] [distance] [permanent] [tag tag]`	Configures a static IP route, where: • *vrf-name*—Specifies the name of the VPN routing/forwarding instance (VRF) for the static route. • *prefix*—Specifies the IP route prefix for the destination. • *mask*—Specifies the prefix mask for the destination. • *next-hop-address*—Specifies the IP address of the next hop that can be used to reach the destination network. • *interface interface-number*—Specifies the network interface type and interface number that can be used to reach the destination network. • **global**—Specifies that the given next hop address is in the non-VRF routing table. • *distance*—Specifies an administrative distance for the route. • **tag** *tag*—Specifies a tag value that can be used as a "match" value for controlling redistribution via route maps. • **permanent**—Specifies that the route will not be removed, even if the interface shuts down.

Verifying a Static Route Using VRF

To verify that the GGSN has established the static VRF route that you configured, use the **show ip route vrf** privileged EXEC command as shown in the following example:

```
Router# show ip route vrf vpn1 static
     172.16.0.0/32 is subnetted, 1 subnets
U       172.16.0.1 [1/0] via 0.0.0.0, Virtual-Access2
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
S       10.100.0.3/32 [1/0] via 10.110.0.13
```

Configuring an OSPF Route Using VRF

To configure an OSPF route using VRF, use the following command beginning in global configuration mode:

Command	Purpose
`Router(config)# router ospf process-id [vrf vrf-name]`	Enables OSPF routing, and enters router configuration mode, where, • *process-id*—Specifies an internally used identification parameter for an OSPF routing process. The *process-id* is locally assigned and can be any positive integer. A unique value is assigned for each OSPF routing process. • **vrf** *vrf-name*—Specifies the name of the VPN routing/forwarding instance.

Configuring an Interface to a PDN Using VRF

To configure a physical interface to the PDN using Fast Ethernet over the Gi interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type slot/port`	Defines a physical interface on the GGSN, where *type* is **fastethernet**, and *slot/port* is the hardware slot and port on the interface.
Step 2	`Router(config-if)# ip route-cache cef`	Enables CEF operation on an interface. **Note** If you are using VRF for VPN access, you must enable CEF switching on the GGSN. If you enable CEF switching at the global configuration level, then it is automatically enabled for each interface unless it has been specifically disabled at the interface.

	Command	Purpose
Step 3	`Router(config-if)# ip vrf forwarding vrf-name`	Associates a VRF with an interface or subinterface. **Note** The *vrf-name* argument should match the name of the VRF that you configured using the **ip vrf** command.
Step 4	`Router(config-if)# ip address ip-address mask [secondary]`	Specifies an IP address for the interface, where: • *ip-address*—Specifies the IP address of the interface in dotted decimal format. • *mask*—Specifies a subnet mask in dotted decimal format. • **secondary**—Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.

Configuring Access to a VPN

After you have completed the prerequisite configuration tasks, you can use one of the following methods to configure access to a VPN:

- Configuring Access to a VPN Without a Tunnel
- Configuring Access to a VPN With a Tunnel

Configuring Access to a VPN Without a Tunnel

If you configure more than one Gi interface to different PDNs, and need to access a VPN off one of those PDNs, then you can configure access to that VPN without configuring an IP tunnel. To configure access to the VPN in this case, you need to configure the **vrf** access point configuration command.

To configure access to a VPN in the GPRS access point list, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gprs access-point-list list-name`	Specifies a name for a new access point list, or references the name of the existing access point list and enters access-point list configuration mode.
Step 2	`Router(config-ap-list)# access-point access-point-index`	Specifies an index number for a new access point definition, or references an existing access point definition, and enters access point configuration mode.

	Command	Purpose
Step 3	`Router(config-access-point)# access-point-name apn-name`	Specifies the network (or domain) name for a PDN that users can access from the GGSN at a defined access point. **Note** The *apn-name* must match the APN that has been provisioned at the MS, HLR, and DNS server.
Step 4	`Router(config-access-point)# access-type real`	Specifies an APN type that corresponds to a physical interface to an external network on the GGSN. This is the default value.
Step 5	`Router(config-access-point)# vrf vrf-name`	Configures VPN routing and forwarding at a GGSN access point and associates the access point with a particular VRF instance.
Step 6	`Router(config-access-point)# exit`	Exits access point configuration mode.

For information about the other access point configuration options, see the "Configuring Other Access Point Options" section on page 6-19.

Configuring Access to a VPN With a Tunnel

If you have only a single Gi interface to a PDN from which you need to access one or more VPNs you can configure an IP tunnel to access those private networks.

To configure access to the VPN in this case, perform the following tasks:

- Configuring the VPN Access Point (Required)
- Configuring the IP Tunnel (Required)

Configuring the VPN Access Point

To configure access to a VPN in the GPRS access point list, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gprs access-point-list list-name`	Specifies a name for a new access point list, or references the name of the existing access point list and enters access-point list configuration mode.
Step 2	`Router(config-ap-list)# access-point access-point-index`	Specifies an index number for a new access point definition, or references an existing access point definition, and enters access point configuration mode.
Step 3	`Router(config-access-point)# access-point name apn-name`	Specifies the access point network ID, which is commonly an Internet domain name. **Note** The *apn-name* must match the APN that has been provisioned at the MS, HLR, and DNS server.

	Command	Purpose
Step 4	`Router(config-access-point)# access-type real`	Specifies an APN type that corresponds to a physical interface to an external network on the GGSN. This is the default value.
Step 5	`Router(config-access-point)# vrf vrf-name`	Configures VPN routing and forwarding at a GGSN access point and associates the access point with a particular VRF instance.
Step 6	`Router(config-access-point)# exit`	Exits access point configuration mode.

For information about the other access point configuration options, see the "Configuring Other Access Point Options" section on page 6-19.

Configuring the IP Tunnel

When you configure a tunnel, you might consider using loopback interfaces as the tunnel endpoints rather than real physical interfaces because loopback interfaces are always up.

To configure an IP tunnel to a private network, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface tunnel number`	Configures a logical tunnel interface number.
Step 2	`Router(config-if)# ip address ip-address mask [secondary]`	Specifies an IP address for the tunnel interface. **Note** This IP address is not used in any other part of the GGSN configuration.
Step 3	`Router(config-if)# tunnel source {ip-address | type number}`	Specifies the IP address (or interface type and port or card number) of the Gi interface to the PDN or a loopback interface.
Step 4	`Router(config-if)# tunnel destination {hostname | ip-address}`	Specifies IP address (or host name) of the private network that you can access from this tunnel.

Configuring Other Access Point Options

This section summarizes the configuration options that you can specify for a GGSN access point.

Some of these options are used in combination with other global router settings to configure the GGSN. Further details about configuring several of these options are discussed in other topics in this chapter and other chapters of this book.

Note Although the Cisco IOS software allows you to configure other access point options on a virtual access point, only the **access-point-name** and **access-type** commands are applicable to a virtual access point.

To configure options for a GGSN access point, use any of the following commands beginning in access-point list configuration mode:

	Command	Purpose
Step 1	`Router(config-ap-list)# access-point access-point-index`	Specifies an index number for a new access point definition, or references an existing access point definition, and enters access point configuration mode.
Step 2	`Router(config-access-point)# access-point-name apn-name`	Specifies the network (or domain) name for a PDN that users can access from the GGSN at a defined access point. **Note** The *apn-name* must match the APN that has been provisioned at the MS, HLR, and DNS server.
Step 3	`Router(config-access-point)# aaa-accounting {enable | disable}`	Enables or disables accounting for a particular access point on the GGSN. **Note** If you have configured a transparent access APN and you want to provide accounting at that APN, you need to configure the **aaa-accounting enable** command at the APN.

	Command	Purpose
Step 4	`Router(config-access-point)# aaa-group {authentication | accounting} server-group`	Specifies a default AAA server group and assigns the type of AAA services to be supported by the server group for a particular access point on the GGSN, where: • **authentication**—Assigns the selected server group for authentication services on the APN. • **accounting**—Assigns the selected server group for accounting services on the APN. • *server-group*—Specifies the name of a AAA server group to be used for AAA services on the APN. **Note** The name of the AAA server group that you specify must correspond to a server group that you configure using the **aaa group server** command.
Step 5	`Router(config-access-point)# access-type {virtual | real}`	(Optional) Specifies the type of access point. The available options are: • **virtual**—APN type that is not associated with any specific physical target network. • **real**—APN type that corresponds to a physical interface to an external network on the GGSN. This is the default value.
Step 6	`Router(config-access-point)# access-mode {transparent | non-transparent}`	(Optional) Specifies whether the GGSN requests user authentication at the access point to a PDN. The available options are: • **transparent**—No security authorization or authentication is requested by the GGSN for this access point. This is the default value. • **non-transparent**—GGSN acts as a proxy for authenticating.
Step 7	`Router(config-access-point)# access-violation deactivate-pdp-context}`	(Optional) Specifies that a user's session be ended and the user packets discarded when a user attempts unauthorized access to a PDN through an access point.
Step 8	`Router(config-access-point)# aggregate {auto | ip-network-prefix{/mask-bit-length | ip-mask}}`	(Optional) Configures the GGSN to create an aggregate route in its IP routing table when receiving PDP requests from MSs on the specified network through a particular access point on the GGSN.
Step 9	`Router(config-access-point)# anonymous user username [password]`	(Optional) Configures anonymous user access at an access point.
Step 10	`Router(config-access-point)# block-foreign-ms`	(Optional) Restricts GGSN access based on the mobile user's home PLMN.

	Command	Purpose
Step 4	`Router(config-access-point)# aaa-group {authentication | accounting} server-group`	Specifies a default AAA server group and assigns the type of AAA services to be supported by the server group for a particular access point on the GGSN, where: • **authentication**—Assigns the selected server group for authentication services on the APN. • **accounting**—Assigns the selected server group for accounting services on the APN. • *server-group*—Specifies the name of a AAA server group to be used for AAA services on the APN. **Note** The name of the AAA server group that you specify must correspond to a server group that you configure using the **aaa group server** command.
Step 5	`Router(config-access-point)# access-type {virtual | real}`	(Optional) Specifies the type of access point. The available options are: • **virtual**—APN type that is not associated with any specific physical target network. • **real**—APN type that corresponds to a physical interface to an external network on the GGSN. This is the default value.
Step 6	`Router(config-access-point)# access-mode {transparent | non-transparent}`	(Optional) Specifies whether the GGSN requests user authentication at the access point to a PDN. The available options are: • **transparent**—No security authorization or authentication is requested by the GGSN for this access point. This is the default value. • **non-transparent**—GGSN acts as a proxy for authenticating.
Step 7	`Router(config-access-point)# access-violation deactivate-pdp-context}`	(Optional) Specifies that a user's session be ended and the user packets discarded when a user attempts unauthorized access to a PDN through an access point.
Step 8	`Router(config-access-point)# aggregate {auto | ip-network-prefix{/mask-bit-length | ip-mask}}`	(Optional) Configures the GGSN to create an aggregate route in its IP routing table when receiving PDP requests from MSs on the specified network through a particular access point on the GGSN.
Step 9	`Router(config-access-point)# anonymous user username [password]`	(Optional) Configures anonymous user access at an access point.
Step 10	`Router(config-access-point)# block-foreign-ms`	(Optional) Restricts GGSN access based on the mobile user's home PLMN.

	Command	Purpose
Step 11	`Router(config-access-point)# dhcp-gateway-address ip-address`	(Optional) Specifies a DHCP gateway to handle DHCP requests for mobile station (MS) users entering a particular PDN access point.
Step 12	`Router(config-access-point)# dhcp-server {ip-address} [ip-address] [vrf]`	(Optional) Specifies a primary (and backup) DHCP server to allocate IP addresses to MS users entering a particular PDN access point.
Step 13	`Router(config-access-point)# gtp response-message wait-accounting`	(Optional) Configures the GGSN to wait for a RADIUS accounting response before sending a create PDP context response to the SGSN.
Step 14	`Router(config-access-point)# ip-access-group access-list-number {in | out}`	(Optional) Specifies access permissions between an MS and a PDN through the GGSN at a particular access point, where *access-list-number* specifies the IP access list definition to be used at the access point. The available options are: • **in**—Applies the IP access list definition from the PDN to the MS. • **out**—Applies the IP access list definition from the MS to the PDN.
Step 15	`Router(config-access-point)# ip-address-pool {dhcp-proxy-client | radius-client | disable}`	(Optional) Specifies a dynamic address allocation method using IP address pools for the current access point. The available options are: • **dhcp-proxy-client**—DHCP server provides the IP address pool. • **radius-client**—RADIUS server provides the IP address pool. • **disable**—Turns off dynamic address allocation. **Note** If you are using a dynamic address allocation method, then you must configure this command according to the appropriate IP address pool source.
Step 16	`Router(config-access-point)# msisdn suppression [value]`	(Optional) Specifies that the GGSN overrides the MSISDN number with a pre-configured value in its authentication requests to a RADIUS server.
Step 17	`Router(config-access-point)# network-request-activation`	(Optional) Enables an access point for network-initiated PDP requests through a VPN.

	Command	Purpose
Step 18	`Router(config-access-point)# ppp-regeneration [max-session number] [setup-time seconds]`	(Optional) Enables an access point to support PPP regeneration, where • **max-session** *number*—Specifies the maximum number of PPP regenerated sessions allowed at the access point. The default value is device dependent and is determined by the maximum number of IDBs that can be supported by the router. • **setup-time** *seconds*—Specifies the maximum amount of time (between 1 and 65535 seconds) within which a PPP regenerated session must be established. The default value is 60 seconds.
Step 19	`Router(config-access-point) redirect intermobile ip ip address`	(Optional) Specifies that mobile-to-mobile traffic be redirected.
Step 20	`Router(config-access-point) security verify {source | destination}`	Specifies that the GGSN verify the source or destination address in TPDUs received from a Gn interface.
Step 21	`Router(config-access-point)# session idle-time number`	(Optional) Specifies the time that the GGSN waits before purging idle mobile sessions for the current access point.
Step 22	`Router(config-access-point)# subscription-required`	(Optional) Specifies that the GGSN checks the value of the selection mode in a PDP context request to determine if a subscription is required to access a PDN through the access point.
Step 23	`Router(config-access-point)# vrf vrf-name`	(Optional) Configures VPN routing and forwarding at a GGSN access point and associates the access point with a particular VRF instance.

Verifying the Access Point Configuration

This section describes how to verify that you have successfully configured access points on the GGSN, and includes the following tasks:

- Verifying the GGSN Configuration, page 6-24
- Verifying Reachability of the Network Through the Access Point, page 6-27

Verifying the GGSN Configuration

To verify that you have properly configured access points on the GGSN, use the **show running-config** command and the **show gprs access-point** commands.

Note The **gprs access-point-list** command first appears in the output of the **show running-config** command under the virtual template interface, which indicates that the gprs access point list has been configured and is associated with the virtual template. To verify your configuration of specific access points within the gprs access point list, look further down in the show output where the **gprs access-point-list** command appears again followed by the individual access point configurations.

Step 1 From global configuration mode, use the **show running-config** command as shown in the following example. Verify that the **gprs access-point-list** command appears under the virtual template interface, and verify the individual access point configurations within the **gprs access-point-list** section of the output as shown in bold:

```
ggsn# 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
!
hostname ggsn
!
no logging buffered
logging rate-limit console 10 except errors
!
aaa new-model
aaa group server radius foo
 server 172.18.43.7 auth-port 1645 acct-port 1646
aaa authentication ppp foo group foo
aaa authorization network foo group foo
aaa accounting network foo start-stop group foo

!
ip subnet-zero
!
!
ip cef
no ip dhcp-client network-discovery
!
!
interface Loopback1
 ip address 10.2.3.4 255.255.255.255
!
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
 duplex half
```

```
!
interface Ethernet1/1
 description Gi interface to gprs.cisco.com
 ip address 10.9.9.4 255.255.255.0
 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
!
ip default-gateway 172.18.43.161
ip kerberos source-interface any
ip classless
ip route 10.7.7.0 255.255.255.0 10.8.8.2
ip route 10.102.82.0 255.255.255.0 172.18.43.161
ip route 192.168.0.0 255.255.0.0 172.18.43.161
ip route 172.18.0.0 255.255.0.0 172.18.43.161
no ip http server
!
. . .
!
gprs access-point-list gprs
!
  access-point 1
   access-point-name gprs.cisco.com
   access-mode non-transparent
   aaa-group authentication foo
   network-request-activation
   exit
!
  access-point 2
   access-point-name gprt.cisco.com
   exit
!
  access-point 3
   access-point-name gpru.cisco.com
   ip-address-pool radius-client
   access-mode non-transparent
   aaa-group authentication foo
   exit
!
gprs maximum-pdp-context-allowed 90000
gprs gtp path-echo-interval 0
gprs default charging-gateway 10.15.15.1
!
. . .
!
```

```
radius-server host 172.18.43.7 auth-port 1645 acct-port 1646 non-standard
radius-server retransmit 3
radius-server key 7 12150415
call rsvp-sync
!
no mgcp timer receive-rtcp
!
mgcp profile default
!
!
gatekeeper
 shutdown
end
```

Step 2 To view the configuration of a specific access point on the GGSN in further detail, use the **show gprs access-point** command and specify the index number of the access point, as shown in the following example:

```
ggsn# show gprs access-point 2
    apn_index 2          apn_name = gprt.cisco.com
    apn_mode: transparent
    apn-type: Real
    accounting: Disable
    wait_accounting: Disable
    dynamic_address_pool: not configured
    apn_dhcp_server: 0.0.0.0
    apn_dhcp_gateway_addr: 0.0.0.0
    apn_authentication_server_group:
    apn_accounting_server_group:
    apn_username: , apn_password:
    subscribe_required: No
    deactivate_pdp_context_on violation: No
    network_activation_allowed: No
    Block Foreign-MS Mode: Disable
    VPN: Disable
    GPRS vaccess interface: Virtual-Access1
    number of ip_address_allocated 0

    Total number of PDP in this APN :1

    aggregate:
    In APN:    Disable

    In Global: Disable
```

Step 3 To view a summary of every access point that is configured on the GGSN, use the **show gprs access-point all** command as shown in the following example:

```
Router# show gprs access-point all

There are 3 Access-Points configured

Index   Mode            Access-type    AccessPointName      VRF Name
-----------------------------------------------------------------------
1       non-transparent    Real       gprs.cisco.com
-----------------------------------------------------------------------
2       transparent        Real       gprt.cisco.com
-----------------------------------------------------------------------
3       non-transparent    Real       gpru.cisco.com
-----------------------------------------------------------------------
```

Verifying Reachability of the Network Through the Access Point

The following procedure provides a basic methodology for verifying reachability from the MS to the destination network.

Note There are many factors that can affect whether or not you can successfully reach the destination network. Although this procedure does not attempt to fully address those factors, it is important for you to be aware that your particular configuration of the APN, IP routing, and physical connectivity of the GGSN, can affect end-to-end connectivity between a host and an MS.

To verify that you can reach the network from the MS, perform the following steps:

Step 1 From the MS (for example, using a handset), create a PDP context with the GGSN by specifying the APN to which you want to connect.

In this example, you specify the APN *gprt.cisco.com*.

Step 2 From global configuration mode on the GGSN, use the **show gprs access-point** command and verify the number of created network PDP contexts (in the Total number of PDP in this APN output field).

The following example shows one successful PDP context request:

```
ggsn# show gprs access-point 2
    apn_index 2          apn_name = gprt.cisco.com
    apn_mode: transparent
    apn-type: Real
    accounting: Disable
    wait_accounting: Disable
    dynamic_address_pool: not configured
    apn_dhcp_server: 0.0.0.0
    apn_dhcp_gateway_addr: 0.0.0.0
    apn_authentication_server_group:
    apn_accounting_server_group:
    apn_username: , apn_password:
    subscribe_required: No
    deactivate_pdp_context_on violation: Yes
    network_activation_allowed: No
    Block Foreign-MS Mode: Disable
    VPN: Disable
    GPRS vaccess interface: Virtual-Access1
    number of ip_address_allocated 0

    Total number of PDP in this APN :1

    aggregate:
    In APN:    Disable

    In Global: Disable
```

Step 3 To test further, generate traffic to the network. To do this, you can use the **ping** command from a handset, or a laptop connected to the handset, to a host on the destination network, as shown in the following example:

```
ping 192.168.12.5
```

Note To avoid possible DNS configuration issues, try to use the IP address (rather than host name) of a host that you expect to be reachable within the destination network. For this test to work, the IP address of the host that you select must be able to be properly routed by the GGSN.

In addition, the APN configuration and physical connectivity to the destination network through a Gi interface must be established. For example, if the host to be reached is in a VPN, the APN must be properly configured to provide access to the VPN.

Step 4 After you have begun to generate traffic over the PDP context, use the **show gprs gtp pdp-context tid** command to see detailed statistics including send and receive byte and packet counts.

Tip To find the TID for a particular PDP context on an APN, use the **show gprs gtp pdp-context access-point** command.

The following example shows sample output for a PDP context for TID 81726354453647FA:

```
ggsn# show gprs gtp pdp-context tid 81726354453647FA

TID               MS Addr         Source  SGSN Addr       APN
81726354453647FA 10.2.2.1         Static  172.16.44.1     gprt.cisco.com

    current time :Dec 06 2001 13:15:34
    user_name (IMSI): 18273645546374     MS address: 10.2.2.1
    MS International PSTN/ISDN Number (MSISDN): 243926901
    sgsn_addr_signal: 172.16.44.1        ggsn_addr_signal: 10.30.30.1
    signal_sequence:  7                  seq_tpdu_up:      0
    seq_tpdu_down:    5380
    upstream_signal_flow:   371          upstream_data_flow:   372
    downstream_signal_flow: 1            downstream_data_flow: 1
    RAupdate_flow:          0
    pdp_create_time:   Dec 06 2001 09:54:43
    last_access_time:  Dec 06 2001 13:15:21
    mnrgflag:          0                 tos mask map: 00
    gtp pdp idle time: 72
    gprs qos_req: 091101               canonical Qos class(req.): 01
    gprs qos_neg: 25131F               canonical Qos class(neg.): 01
    effective bandwidth: 0.0
    rcv_pkt_count:       10026         rcv_byte_count:    1824732
    send_pkt_count:      5380          send_byte_count:   4207160
    cef_up_pkt:          10026         cef_up_byte:       1824732
    cef_down_pkt:        5380          cef_down_byte:     4207160
    cef_drop:            0
    charging_id:         12321224
    pdp reference count: 2
    ntwk_init_pdp:       0
```

Configuring Access to External Support Servers

You can configure the GGSN to access external support servers to provide services for dynamic IP addressing of MSs using the Dynamic Host Configuration Protocol (DHCP) or using Remote Authentication Dial-In User Service (RADIUS). You can also configure RADIUS services on the GGSN to provide security, such as authentication of users accessing a network at an APN.

The GGSN allows you to configure access to DHCP and RADIUS servers globally for all access points, or to specific servers for a particular access point. For more information about configuring DHCP on the GGSN, see the "Configuring DHCP on the GGSN" chapter. For more information about configuring RADIUS on the GGSN, see the "Configuring Security on the GGSN" chapter.

Configuring Virtual APN Access on the GGSN

This section includes the following topics:

For a sample configuration, see the "Virtual APN Configuration Example" section on page 6-51.

Overview of the Virtual APN Feature

GGSN Release 3.0 and later supports virtual APN access from the PLMN using the virtual access point type on the GGSN. The virtual APN feature on the GGSN allows multiple users to access different physical target networks through a shared APN access point on the GGSN.

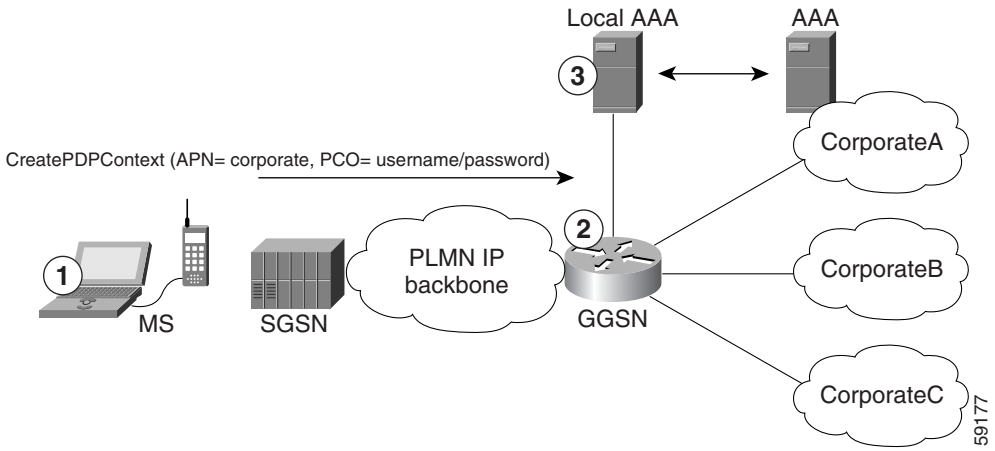
In a GPRS/UMTS network, the user APN information must be configured at several of the GPRS/UMTS network entities, such as the HLR and DNS server. In the HLR, the user subscription data associates the IMSI (unique per user) with each APN that the IMSI is allowed to access. At the DNS server, APNs are correlated to the GGSN IP address. If DHCP or RADIUS servers are in use, the APN configuration can extend to those servers too.

The virtual APN feature reduces the amount of APN provisioning required by consolidating access to all real APNs through a single virtual APN at the GGSN. Therefore, only the virtual APN needs to be provisioned at the HLR and DNS server, instead of each of the real APNs to be reached. The GGSN also must be configured for the virtual APN.

The Cisco Systems GGSN software determines the ultimate target network for the session by receiving the create PDP context request at the virtual access point and extracting the domain name to direct the packet to the appropriate real APN. The real APN is the actual destination network.

Figure 6-1 shows how the GGSN supports a create PDP context request from an MS processed through a virtual APN on the GGSN.

Figure 6-1 Virtual APN PDP Context Activation on the GGSN



1.	At the MS, the user connects to the network with a username in the form of login@domain, such as ciscouser@CorporateA.com. The SGSN sends a create PDP context request to the GGSN using the virtual APN of "corporate." The create PDP context also includes the username in login@domain format in the protocol configuration option (PCO) information element.
2.	The GGSN extracts the domain from the information in the PCO, which corresponds to the real target network on the GGSN. In this example, the GGSN finds CorporateA.com as the domain and directs the session to the appropriate real APN for the target network. In this case, the real APN is corporateA.com. The GGSN uses the complete username to do authentication.
3.	The local or corporate AAA server is selected based on the domain part of the username, which is CorporateA.com in this case.

Benefits of the Virtual APN Feature

The virtual APN feature provides the following benefits:

- Simplifies provisioning of APN information at the HLR and DNS servers.
- Improves scalability for support of large numbers of corporate networks, ISPs, and services.
- Increases flexibility of access point selection.
- Eases deployment of new APNs and services.

Restrictions of the virtual APN Feature

The virtual APN feature has the following restriction:

- S-CDRs and G-CDRs do not include the domain information.

Virtual APN Configuration Task List

To configure the GGSN to support virtual APN access, you must configure one or more virtual access points. You also need to configure the real access points that provide the information needed to connect to the physical networks of the external PDNs or VPNs.

In addition to the configuring the GGSN, you must also ensure proper provisioning of other GPRS/UMTS network entities as appropriate to successfully implement the virtual APN feature on the GPRS/UMTS network.

To configure virtual APN access on the GGSN, perform the following tasks:

- Configuring Virtual Access Points on the GGSN, page 6-31 (Required)
- Configuring Real Access Points on the GGSN, page 6-10 (Required)
  - PDN Access Configuration Task List, page 6-10
  - VPN Access Using VRF Configuration Task List, page 6-12

For a sample configuration, see the "Virtual APN Configuration Example" section on page 6-51.

Configuring Virtual Access Points on the GGSN

Use virtual access point types to consolidate access to multiple real target networks on the GGSN. The GGSN always uses real access points to reach an external network, so virtual access points are used in combination with real access points on the GGSN.

You can configure multiple virtual access points on the GGSN. Multiple virtual access points can be used to access the same real networks. One virtual access point can be used to access different real networks.

Note Be sure that you provision the HLR and configure the DNS server to properly correspond to the virtual APN domains that you have configured on the GGSN. For more information, see the "Configuring Other GPRS/UMTS Network Entities With the Virtual APN" section on page 6-32.

To configure a virtual access point on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gprs access-point-list list-name`	Specifies a name for a new access-point list, or references the name of the existing access-point list and enters access-point list configuration mode.
Step 2	`Router(config-ap-list)# access-point access-point-index`	Specifies an index number for a new access point definition, or references an existing access point definition, and enters access point configuration mode.

	Command	Purpose
Step 3	`Router(config-access-point)# access-point-name apn-name`	Specifies the network (or domain) name for a PDN that users can access from the GGSN at a defined access point. **Note** The *apn-name* must match the APN that has been provisioned at the MS, HLR, and DNS server.
Step 4	`Router (config-access-point)# access-type virtual`	Specifies an APN type that is not associated with any specific physical target network on the GGSN. The default access type is real.

Note Although the Cisco IOS software allows you to configure other access point options on a virtual access point, no other access point options are applicable if they are configured.

Configuring Other GPRS/UMTS Network Entities With the Virtual APN

When you configure the GGSN to support virtual APN access, be sure that you also meet any necessary requirements to properly configure other GPRS/UMTS network entities to support the virtual APN implementation.

The following GPRS/UMTS network entities might also require provisioning to properly implement virtual APN support:

- DHCP server—Requires configuration of the real APNs.
- DNS server—The DNS server that the SGSN uses to resolve the address of the GGSN must identify the virtual APN with the IP address of the GTP virtual template on the GGSN. If GTP SLB is implemented, then the virtual APN should be associated with the IP address of the GTP load balancing virtual server instance on the SLB router.
- HLR—Requires the name of the virtual APN in subscription data, as allowable for subscribed users.
- RADIUS server—Requires configuration of the real APNs.
- SGSN—Requires the name of the virtual APN as the default APN (as desired) when the APN is not provided in user subscription data.

Verifying the Virtual APN Configuration

This section describes how to verify that you have successfully configured virtual APN support on the GGSN, and includes the following tasks:

- Verifying the GGSN Configuration, page 6-33
- Verifying Reachability of the Network Through the Virtual Access Point, page 6-36

Verifying the GGSN Configuration

To verify that you have properly configured access points on the GGSN, use the **show running-config** command and the **show gprs access-point** commands.

Note The **gprs access-point-list** command first appears in the output of the **show running-config** command under the virtual template interface, which indicates that the gprs access point list has been configured and is associated with the virtual template. To verify your configuration of specific access points within the gprs access point list, look further down in the show output where the **gprs access-point-list** command appears again followed by the individual access point configurations.

Step 1 From privileged EXEC mode, use the **show running-config** command as shown in the following example. Verify the interface configuration and virtual and real access points as shown by the arrows:

```
ggsn# 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
!
! Enable the router for GGSN services
!
service gprs ggsn
!
hostname ggsn
!
no logging buffered
logging rate-limit console 10 except errors
aaa new-model
aaa group server radius foo
 server 172.18.43.7 auth-port 1645 acct-port 1646
aaa authentication ppp foo group foo
aaa authorization network foo group foo
aaa accounting network foo start-stop group foo

!
ip subnet-zero
!
!
no ip dhcp-client network-discovery
!
!
interface Loopback1
 ip address 10.2.3.4 255.255.255.255
!
interface FastEthernet0/0
 ip address 172.18.43.174 255.255.255.240
 duplex half
!
interface FastEthernet2/0
 description Gn interface
 ip address 192.168.10.56 255.255.255.0
!
! Define Gi physical interfaces to real networks
!
```

```
interface Ethernet1/0
 description Gi interface to corporatea.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 corporateb.com
 ip address 10.9.9.4 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex half
!
interface Ethernet1/2
 description Gi interface to corporatec.com
 ip address 10.15.15.10 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 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
!
ip default-gateway 172.18.43.161
ip kerberos source-interface any
ip classless
ip route 10.7.7.0 255.255.255.0 10.8.8.2
ip route 10.102.82.0 255.255.255.0 172.18.43.161
ip route 192.168.1.1 255.255.255.255 FastEthernet2/0
ip route 172.18.0.0 255.255.0.0 172.18.43.161
no ip http server
!
gprs access-point-list gprs
!
! Configure a virtual access point called corporate
!
 access-point 1
   access-point-name corporate
   access-type virtual
   exit
!
! Configure three real access points called corporatea.com,
! corporateb.com, and corporatec.com
!
 access-point 2
   access-point-name corporatea.com
   access-mode non-transparent
   aaa-group authentication foo
   exit
!
 access-point 3
   access-point-name corporateb.com
   exit
   !
```

```
 access-point 4
   access-point-name corporatec.com
   access-mode non-transparent
   aaa-group authentication foo
   exit
   !
 !
gprs maximum-pdp-context-allowed 90000
gprs gtp path-echo-interval 0
gprs default charging-gateway 10.15.15.1
!
radius-server host 172.18.43.7 auth-port 1645 acct-port 1646 non-standard
radius-server retransmit 3
radius-server key 7 12150415
call rsvp-sync
!
no mgcp timer receive-rtcp
!
mgcp profile default
!
!
gatekeeper
 shutdown
!
end
```

Step 2 To view the configuration of a specific access point on the GGSN in further detail, use the **show gprs access-point** command and specify the index number of the access point, as shown in the following examples.

The following output shows information about a real access point:

```
ggsn# show gprs access-point 2
    apn_index 2          apn_name = corporatea.com
    apn_mode: non-transparent
    apn-type: Real
    accounting: Disable
    wait_accounting: Disable
    dynamic_address_pool: not configured
    apn_dhcp_server: 0.0.0.0
    apn_dhcp_gateway_addr: 0.0.0.0
    apn_authentication_server_group: foo
    apn_accounting_server_group:
    apn_username: , apn_password:
    subscribe_required: No
    deactivate_pdp_context_on violation: No
    network_activation_allowed: No
    Block Foreign-MS Mode: Disable
    VPN: Disable
    GPRS vaccess interface: Virtual-Access1
    number of ip_address_allocated 0

    Total number of PDP in this APN :1

    aggregate:
    In APN:    Disable

    In Global: Disable
```

The following output shows information about a virtual access point:

```
ggsn# show gprs access-point 1
    apn_index 1          apn_name = corporate
    apn_mode: transparent
    apn-type: Virtual
    accounting: Disable
    wait_accounting: Disable
    dynamic_address_pool: not configured
    apn_dhcp_server: 0.0.0.0
    apn_dhcp_gateway_addr: 0.0.0.0
    apn_authentication_server_group:
    apn_accounting_server_group:
    apn_username: , apn_password:
    subscribe_required: No
    deactivate_pdp_context_on violation: No
    network_activation_allowed: No
    Block Foreign-MS Mode: Disable
    VPN: Disable
    GPRS vaccess interface: Virtual-Access2
    number of ip_address_allocated 0

    Total number of PDP in this APN :0

    aggregate:
    In APN:    Disable

    In Global: Disable
```

Step 3 To view a summary of every access point that is configured on the GGSN, use the **show gprs access-point all** command as shown in the following example:

```
ggsn# show gprs access-point all

There are 4 Access-Points configured

Index   Mode            Access-type    AccessPointName      VRF Name
-----------------------------------------------------------------------
1       transparent        Virtual    corporate
-----------------------------------------------------------------------
2       non-transparent    Real       corporatea.com
-----------------------------------------------------------------------
3       transparent        Real       corporateb.com
-----------------------------------------------------------------------
4       non-transparent    Real       corporatec.com
-----------------------------------------------------------------------
```

Verifying Reachability of the Network Through the Virtual Access Point

To verify reachability of the real destination network through the virtual access point, you can use the same procedure described in the "Verifying Reachability of the Network Through the Access Point" section on page 6-27.

In addition, you should meet the following guidelines for virtual access point testing:

- When you initiate PDP context activation at the MS, be sure that the username that you specify (in the form of login@domain in the create PDP context request) corresponds to a real APN that you have configured on the GGSN.
- When you generate traffic to the network, be sure to select a host on one of the real destination networks that is configured for APN support on the GGSN.

Configuring Network-Initiated PDP Context Support on the GGSN

This section includes the following topics:

For a sample configuration, see the "Network-Initiated PDP Request Configuration Example" section on page 6-54.

Overview of Network-Initiated PDP Context Support

In GPRS Release 1.4 and earlier, the GGSN only supports creation of PDP contexts that are originated by an MS. GGSN Release 3.0 and later supports network-initiated PDP contexts for statically configured IP addresses. This means that the GGSN supports a process for creating PDP contexts initiated by an external IP network.

When the GGSN receives a PDU destined for an MS from the IP network, it verifies whether a PDP context is already established for that MS on the GGSN. If the MS does not have an existing PDP context on the GGSN, then the GGSN issues a Send Routing Information request to the home location register (HLR). The GGSN uses a GSN that provides the necessary GTP-to-Mobile Application Part (MAP) conversion to communicate with the HLR. If the HLR determines that the Send Routing Information request can be served, it sends the GGSN the address of the SGSN (through the protocol-converting GSN) that is currently serving that MS. The GGSN sends a PDU Notification Request to the SGSN serving the MS, and the SGSN requests that the MS establish the PDP context with the GGSN.

Restrictions

The GGSN supports creation of network-initiated PDP contexts with the following restrictions:

- IP addresses corresponding to the International Mobile Subscriber Identity (IMSI) of an MS must be statically configured on the GGSN using the **gprs ni-pdp ip-imsi single** command.
- If you are implementing VPN access through a VRF at the access point, you must configure the access point for VRF *before* you configure the IP to IMSI address mappings using the **gprs ni-pdp ip-imsi single** global configuration command. If you configure the **gprs ni-pdp ip-imsi single** command before you configure VRF at the access point, then the addresses that you specify become part of the global routing table and *not* the VRF routing table.

Network-Initiated PDP Context Configuration Task List

The GGSN supports network-initiated PDP contexts for both VPN and non-VPN networks. However, access through a VPN is preferable for greater flexibility in IP addressing and better control over security and other functions at the GGSN access point.

To configure network-initiated PDP context support on the GGSN through a VPN, perform the following tasks:

- Configuring Network-Initiated PDP Context Support at an APN, page 6-38 (Required)
- Specifying the GSN for GTP-MAP Protocol Conversion, page 6-39 (Required)
- Configuring the Static IP Address Mapping to IMSI, page 6-39 (Required)
- Configuring Other Network-Initiated PDP Options, page 6-40 (Optional)

To verify your configuration, see the "Verifying the Network-Initiated PDP Context Configuration" section on page 6-41.

For a sample configuration, see the "Network-Initiated PDP Request Configuration Example" section on page 6-54.

Configuring Network-Initiated PDP Context Support at an APN

To support network-initiated PDP context activation on the GGSN at a specific APN, you must enable network request activation at the access point.

The GGSN supports network-initiated PDP contexts at multiple VPNs. To do this, you must create an access point for each VPN that you want to support and you must configure VRF at the APN. In addition to configuring VRF at the APN, other tasks are required to complete the VRF configuration. For more information about configuring VRF support on the GGSN, see the "VPN Access Using VRF Configuration Task List" section on page 6-12.

To configure network-initiated PDP context support at an APN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gprs access-point-list list-name`	Specifies a name for a new access point list, or references the name of the existing access point list and enters access-point list configuration mode.
Step 2	`Router(config-ap-list)# access-point access-point-index`	Specifies an index number for a new access point definition, or references an existing access point definition, and enters access point configuration mode. **Note** The *access-point-index* that you specify in this command must correspond to the *apn-index* in the **gprs ni-pdp ip-imsi single** command.
Step 3	`Router(config-access-point)# access-point-name apn-name`	Specifies the network (or domain) name for a PDN that users can access from the GGSN at a defined access point. **Note** The *apn-name* must match the APN that has been provisioned at the MS, HLR, and DNS server.

	Command	Purpose
Step 4	`Router(config-access-point)# network-request-activation`	Enables an access point for network-initiated PDP requests.
Step 5	`Router(config-access-point)# vrf vrf-name`	(Optional) Configures VPN routing and forwarding at a GGSN access point and associates the access point with a particular VRF instance.

For information about other access point configuration options, see the "Configuring the GPRS Access Point List on the GGSN" section on page 6-9.

Specifying the GSN for GTP-MAP Protocol Conversion

To specify the address of the GSN for GTP-MAP protocol conversion, use the following command in global configuration mode:

Command	Purpose
`Router(config)# gprs default map-converting-gsn {ip-address | hostname} [ip-address | hostname]`	Specifies the IP address or host name of the primary (and backup) GSN to communicate with the HLR in sending and receiving MAP messages.

Configuring the Static IP Address Mapping to IMSI

The GGSN supports network-initiated PDP context requests from both a VPN or other intranet using statically configured address mappings only.

When you configure the static IP address mapping to IMSI, you must specify the proper APN number where you have enabled the **network-request-activation** command.

To configure the static IP address mapping to the IMSI of an MS, use the following command in global configuration mode:

Command	Purpose
`Router(config)# gprs ni-pdp ip-imsi single apn-index ip-address imsi`	Specifies a static IP address to IMSI mapping for a single MS for network-initiated PDP requests from a particular APN, with the following values: • *apn-index*—Specifies the access-point where you have have enabled network-initiated PDP context support using the **network-request-activation** command. • *ip-address*—Specifies the static IP address of that corresponds to the PDP address in the request coming from the APN. • *imsi*—Specifies the international mobile subscriber identity of the MS that you want to map to the configured *ip-address*. Reissue this command for each MS that you want to support, using a different IP address and IMSI value.

Configuring Other Network-Initiated PDP Options

To configure other network-initiated PDP context options on the GGSN, use the following commands beginning in global configuration mode:

Command	Purpose
`Router(config)# gprs ni-pdp pdp-buffer number`	(Optional) Specifies the maximum size of the GGSN buffer to be used for each network-initiated PDP request. The default value is 2000 bytes.
`Router(config)# gprs ni-pdp percentage percentage-number`	(Optional) Specifies the maximum percentage of PDP contexts on the GGSN that can be network-initiated. The default value is 10 percent.
`Router(config)# gprs ni-pdp discard-period number`	(Optional) Specifies the amount of time that the GGSN waits, after an unsuccessful network-initiated PDP delivery attempt, before discarding subsequent PDP PDUs received on the Gi interface. The default value is 300 seconds (5 minutes).
`Router(config)# gprs ni-pdp cache-timeout number`	(Optional) Specifies the maximum amount of time that an SGSN address is cached by the GGSN. The default value is 600 seconds (10 minutes).

Verifying the Network-Initiated PDP Context Configuration

This section describes how to verify that you have sucessfully configured the GGSN for network-initiated PDP context support, and includes the following tasks:

- Verifying the GGSN Configuration, page 6-41
- Verifying Reachability of the MS Using Network-Initiated PDP Request, page 6-44

Verifying the GGSN Configuration

To verify that you have properly configured the GGSN for network-initiated PDP context support, use the **show running-config** and **show gprs access-point** commands.

Step 1 From privileged EXEC mode, use the **show running-config** command as shown in the following example. Verify the access point and global configuration values as shown in bold:

```
ggsn# 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
!
hostname ggsn
!
no logging buffered
logging rate-limit console 10 except errors
aaa new-model
aaa group server radius foo
 server 172.18.43.7 auth-port 1645 acct-port 1646
aaa authentication ppp foo group foo
aaa authorization network foo group foo
aaa accounting exec default start-stop group foo
aaa accounting network foo start-stop group foo
!
ip vrf vpn1
 rd 100:1
!
ip subnet-zero
!
ip cef
!
no ip dhcp-client network-discovery
!
!
interface Loopback1
 ip address 10.2.3.4 255.255.255.255
!
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
 ip vrf forwarding vpn1
 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
!
ip default-gateway 172.18.43.161
ip kerberos source-interface any
ip classless
ip route 10.7.7.0 255.255.255.0 10.8.8.2
ip route 10.102.82.0 255.255.255.0 172.18.43.161
ip route 192.168.0.0 255.255.0.0 172.18.43.161
ip route 172.18.0.0 255.255.0.0 172.18.43.161
no ip http server
!
. . .
!
```

For network-initiated PDP context support at a VPN, verify that you have enabled network-initiated PDP context support at the APN and have properly configured the APNs for VPN access as shown in bold:

```
!
. . .
gprs access-point-list gprs
!
  access-point 1
   access-point-name gprs.cisco.com
   access-mode non-transparent
   aaa-group authentication foo
   network-request-activation
   exit
!
  access-point 2
   access-point-name gprt.cisco.com
   network-request-activation
   vrf vpn1
   exit
   !
  access-point 3
   access-point-name gpru.cisco.com
   access-mode non-transparent
   aaa-group authentication foo
   exit
```

```
   !
gprs maximum-pdp-context-allowed 90000
gprs gtp path-echo-interval 0
gprs default charging-gateway 10.15.15.1
gprs gtp ip udp ignore checksum

!
. . .
!
```

Verify that you have configured the protocol-converting SGSN and configured the IP address-to-IMSI mappings for each of the MSs that you want to support, as shown in bold:

```
!
. . .

gprs default map-converting-gsn 10.7.7.1
gprs ni-pdp ip-imsi single 1 10.100.1.1 11111111111111F1
gprs ni-pdp ip-imsi single 2 172.31.1.2 11111111111111F2
gprs ni-pdp ip-imsi single 2 172.31.1.3 11111111111111F3
!
!
radius-server host 172.18.43.7 auth-port 1645 acct-port 1646 non-standard
radius-server retransmit 3
radius-server key 7 12150415
call rsvp-sync
!
no mgcp timer receive-rtcp
!
mgcp profile default
!
!
gatekeeper
 shutdown
!
end
```

Step 2 From privileged EXEC mode, use the **show gprs access-point** command and verify that the network_activation_allowed output field contains the value Yes, as shown in the following example:

```
ggsn# show gprs access-point 1
    apn_index 1          apn_name = gprs.cisco.com
    apn_mode: non-transparent
    apn-type: Real
    accounting: Disable
    wait_accounting: Disable
    dynamic_address_pool: not configured
    apn_dhcp_server: 0.0.0.0
    apn_dhcp_gateway_addr: 0.0.0.0
    apn_authentication_server_group: foo
    apn_accounting_server_group:
    apn_username: , apn_password:
    subscribe_required: No
    deactivate_pdp_context_on violation: No
    network_activation_allowed: Yes
    Block Foreign-MS Mode: Disable
    VPN: Disable
    GPRS vaccess interface: Virtual-Access1
    number of ip_address_allocated 0
```

```
    Total number of PDP in this APN :0

    aggregate:
    In APN:    Disable

    In Global: Disable
```

Verifying Reachability of the MS Using Network-Initiated PDP Request

To verify that you can reach the MS from the PDN, perform the following steps:

Step 1 From the PDN side of the IP network, generate traffic to the MS. To do this, you can use the **ping** command with the IP address of the MS.

In the configuration example shown in Figure 6-3, you could issue **ping 10.100.1.1**, **ping 172.31.1.2**, or **ping 172.31.1.3**.

Step 2 From privileged EXEC mode on the GGSN, use the **show gprs gtp statistics** command and verify the number of rejected and created network PDP contexts (in the ntwk_init_pdp_act_rej and total ntwkInit created pdp output fields).

The following example shows 1 successful network-initiated PDP context:

```
ggsn# show gprs gtp statistics
GPRS GTP Statistics:
  version_not_support    0              msg_too_short              0
  unknown_msg            0              unexpected_sig_msg         1
  unexpected_data_msg    0              mandatory_ie_missing       0
  mandatory_ie_incorrect 0              optional_ie_invalid        0
  ie_unknown             0              ie_out_of_order            0
  ie_unexpected          0              ie_duplicated              0
  optional_ie_incorrect  0              pdp_activation_rejected    0
  path_failure           0              total_dropped              0
  no_resource            0              get_pak_buffer_failure     0
  rcv_signalling_msg     4              snd_signalling_msg         8
  rcv_pdu_msg            0              snd_pdu_msg                1
  rcv_pdu_bytes          0              snd_pdu_bytes              100
  total created_pdp      1              total deleted_pdp          0
  ntwk_init_pdp_act_rej  0              total ntwkInit created pdp 1
```

Step 3 Use the **show gprs gtp pdp-context tid** command and verify that the ntwk_init_pdp output field contains the value 1, as shown in the following example.

Note To find the TID of a PDP context for a particular MS, use the **show gprs gtp pdp-context ms-address** command.

```
GGSN_1# show gprs gtp pdp-context tid 81726354453647F2
TID              MS Addr         Source  SGSN Addr       APN
81726354453647F2 10.100.1.1      Static  10.7.7.1        gprs.cisco.com

    current time :Dec 06 2001 13:15:34
    user_name (IMSI): 182736455463742     MS address: 10.100.1.1
    MS International PSTN/ISDN Number (MSISDN): 21436587214365
    sgsn_addr_signal: 10.7.7.1           ggsn_addr_signal: 10.30.30.1
    signal_sequence:  7                  seq_tpdu_up:      0
    seq_tpdu_down:    5380
    upstream_signal_flow:   371          upstream_data_flow:   372
```

```
downstream_signal_flow: 1            downstream_data_flow: 1
RAupdate_flow:          0
pdp_create_time:   Dec 06 2001 09:54:43
last_access_time:  Dec 06 2001 13:15:21
mnrgflag:          0                tos mask map: 00
gtp pdp idle time: 72
gprs qos_req: 091101               canonical Qos class(req.): 01
gprs qos_neg: 25131F               canonical Qos class(neg.): 01
effective bandwidth: 0.0
rcv_pkt_count:       10026         rcv_byte_count:  1824732
send_pkt_count:      5380          send_byte_count: 4207160
cef_up_pkt:          10026         cef_up_byte:     1824732
cef_down_pkt:        5380          cef_down_byte:   4207160
cef_drop:            0
charging_id:         12321224
pdp reference count: 2
ntwk_init_pdp:       1
```

Blocking Access to the GGSN by Foreign Mobile Stations

This section describes how to restrict access to the GGSN from mobile stations outside of their home PLMN. It includes the following topics:

- Overview of Blocking Foreign Mobile Stations, page 6-45
- Blocking Foreign Mobile Stations Configuration Task List, page 6-45
- Blocking Access by Foreign Mobile Stations Configuration Example, page 6-57

Overview of Blocking Foreign Mobile Stations

The GGSN allows you to block access by mobile stations who are outside of the PLMN. When you enable blocking of foreign mobile stations, the GGSN determines if an MS is inside or outside of the PLMN based on the mobile country code (MCC) and mobile network code (MNC). You must specify the MCC and MNC codes on the GGSN to properly configure the home public land mobile network (HPLMN) values.

When you enable the blocking foreign MS access feature on the access point, then when the GGSN receives a GTP create PDP context request message, the GGSN compares the MCC and MNC in the TID against the home operator codes that you configure on the GGSN. If the MS mobile operator code fails the matching criteria on the GGSN, then the GGSN rejects the create PDP context request message.

Blocking Foreign Mobile Stations Configuration Task List

To implement blocking of foreign mobile stations on the GGSN, you must enable the function and specify the supporting criteria for determining whether an MS is outside of its home PLMN.

To configure blocking of foreign mobile stations on the GGSN, perform the following tasks:

- Configuring the MCC and MNC Values, page 6-46 (Required)
- Enabling Blocking of Foreign Mobile Stations on the GGSN, page 6-46 (Required)
- Verifying the Blocking of Foreign Mobile Stations Configuration, page 6-46

Configuring the MCC and MNC Values

To configure the MCC and MNC values that the GGSN uses to determine if a request is from a roaming MS, use the following command in global configuration mode:

Command	Purpose
`Router(config)# gprs mcc mcc-num mnc mnc-num`	Configures the mobile country code and mobile network node that the GGSN uses to determine whether a create PDP context request is from a foreign MS.

Note The GGSN automatically specifies values of 000 for the MCC and MNC. However, you must configure non-zero values for both the MCC and MNC.

Enabling Blocking of Foreign Mobile Stations on the GGSN

To enable the GGSN to block foreign mobile stations from establishing PDP contexts, use the following command in global configuration mode:

Command	Purpose
`Router(config-access-point)# block-foreign-ms`	Restricts GGSN access based on the mobile user's HPLMN.

Note The MCC and MNC values used to determine whether a request is from a roaming MS must be configured before the GGSN can be enabled to block foreign mobile stations.

Verifying the Blocking of Foreign Mobile Stations Configuration

This section describes how you can verify the blocking of foreign mobile stations configuration on the GGSN. It includes the following topics:

- Verifying Blocking of Foreign Mobile Stations at an Access Point, page 6-46
- Verifying the MCC and MNC Configuration on the GGSN, page 6-47

Verifying Blocking of Foreign Mobile Stations at an Access Point

To verify whether the GGSN is configured to support blocking of foreign mobile stations at a particular access point, use the **show gprs access-point** command. Observe the value of the Block Foreign-MS Mode output field as shown in bold in the following example:

```
Router# show gprs access-point 1
    apn_index 1          apn_name = gprs.corporate.com
    apn_mode: transparent
    apn-type: Real
    accounting: Disable
    wait_accounting: Disable
    dynamic_address_pool: dhcp-proxy-client
```

```
    apn_dhcp_server: 10.99.100.5
    apn_dhcp_gateway_addr: 10.27.1.1
    apn_authentication_server_group: foo
    apn_accounting_server_group: foo1
    apn_username: , apn_password:
    subscribe_required: No
    deactivate_pdp_context_on violation: Yes
    network_activation_allowed: Yes
    Block Foreign-MS Mode: Enable
    VPN: Enable  (VRF Name : vpn1)
    GPRS vaccess interface: Virtual-Access2
    number of ip_address_allocated 0

    Total number of PDP in this APN :0

    aggregate:
    In APN:    auto

    In Global: 30.30.0.0/16
               21.21.0.0/16
```

Verifying the MCC and MNC Configuration on the GGSN

To verify the configuration elements that the GGSN uses as matching criteria to determine whether a request is coming from a foreign mobile station, use the **show gprs gtp parameters** privileged EXEC command. Observe the values of the output fields shown in bold in the following example. The example shows that the GGSN is configured for the USA country code (310) and for the Bell South network code (15):

```
Router# show gprs gtp parameters
    GTP path echo interval                        = 60
    GTP signal max wait time T3_response          = 1
    GTP max retry N3_request                      = 5
    GTP dynamic echo-timer minimum                = 5
    GTP dynamic echo-timer smooth factor          = 2
    GTP buffer size for receiving N3_buffer       = 8192
    GTP max pdp context                           = 45000
    GPRS MCC Code                                 = 310
    GPRS MNC Code                                 = 15
```

Note For a reference table of some of the established MCC and MNC codes, refer to the Appendix of the *Cisco IOS Mobile Wireless Command Reference*.

Controlling Access to the GGSN by MSs with Duplicate IP Addresses

An MS can not have the same IP address as another GPRS/UMTS network entity. You can configure the GGSN to reserve certain IP address ranges for use by the GPRS/UMTS network, and to disallow them from use by an MS.

During a create PDP context request, the GGSN verifies whether the IP address of an MS falls within the specified excluded range. If there is an overlap of the MS IP address with an excluded range, then the PDP context request is rejected. This measure prevents duplicate IP addressing in the network.

You can configure up to 100 IP address ranges. A range can be one or more addresses. However, you can configure only one IP address range per command entry. To exclude a single IP address, you can repeat the IP address in the start-ip and end-ip arguments. IP addresses are 32-bit values.

To reserve IP address ranges for use by the GPRS/UMTS network and block their use by an MS, use the following command in global configuration mode:

Command	Purpose
`Router(config)# gprs ms-address exclude-range start-ip end-ip`	Specifies the IP address ranges used by the GPRS/UMTS network, and thereby excluded from the MS IP address range.

Configuration Examples

This section includes the following configuration examples for configuring different types of network access to the GGSN:

Static Route to SGSN Example

The following example shows how to configure a static route from a physical interface on the GGSN to the SGSN.

Notice the following areas in the GGSN configuration shown in this example:

- FastEthernet0/0 is the physical interface to the SGSN, which is known as the Gn interface.
- In this example, the SGSN is located at IP address 192.168.1.1. Using the **ip route** command, a static route is configured to the SGSN located at 192.168.1.1 from the FastEthernet0/0 interface on the GGSN.

GGSN Configuration

```
! Configure Gn interface on GGSN to communicate with SGSN
!
interface FastEthernet0/0
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 no keepalive
!
ip route 192.168.1.1 255.255.255.255 FastEthernet0/0
```

Note For the SGSN to successfully communicate with the GGSN, the SGSN must configure a static route, or be able to dynamically route to the IP address used by the GGSN virtual template.

Access Point List Configuration Example

The following example shows a portion of the GGSN configuration for a GPRS access point list:

```
!
interface virtual-template 1
 ip unnumber loopback 1
 no ip directed-broadcast
 encapsulation gtp
 gprs access-point-list abc
!
! Defines a GPRS access point list named abc
! with 3 access points
!
gprs access-point-list abc
 access-point 1
  access-point-name gprs.pdn1.com
  ip-address-pool dhcp-proxy-client
  dhcp-server 10.102.100.3
  dhcp-gateway-address 10.30.30.30
  exit
!
 access-point 2
  access-point-name gprs.pdn2.com
  ip-address-pool dhcp-proxy-client
  dhcp-server 10.60.0.1
  dhcp-gateway-address 10.27.27.27
  exit
!
 access-point 3
```

```
  access-point-name www.pdn3.com
  access-mode non-transparent
  dhcp-gateway-address 10.25.25.25
  aaa-group authentication foo
  exit
!
. . .
```

VRF Tunnel Configuration Example

The following example shows a partial configuration for a virtual private network named “vpn1” using VRF:

```
! Configure a VRF routing table
! and define an identifier
!
ip vrf vpn1
 rd 100:1
!
! Enable CEF switching
!
ip cef
!
interface Loopback101
 ip address 10.14.101.1 255.255.255.255
!
! Configure a tunnel interface
! to a private network using VRF
!
interface Tunnel1
 ip vrf forwarding vpn1
 ip address 10.1.101.1 255.255.255.0
 tunnel source 10.14.101.1
 tunnel destination 10.13.101.1
!
! Configure OSPF routing using VRF
!
router ospf 101 vrf vpn1
log-adjacency-changes
redistribute static subnets
network 10.1.101.0 0.0.0.255 area 0
!
! Configure VRF at the access point
!
gprs access-point-list gprs
 access-point 1
  access-point-name gprs.cisco.com
  vrf vpn1
  exit
```

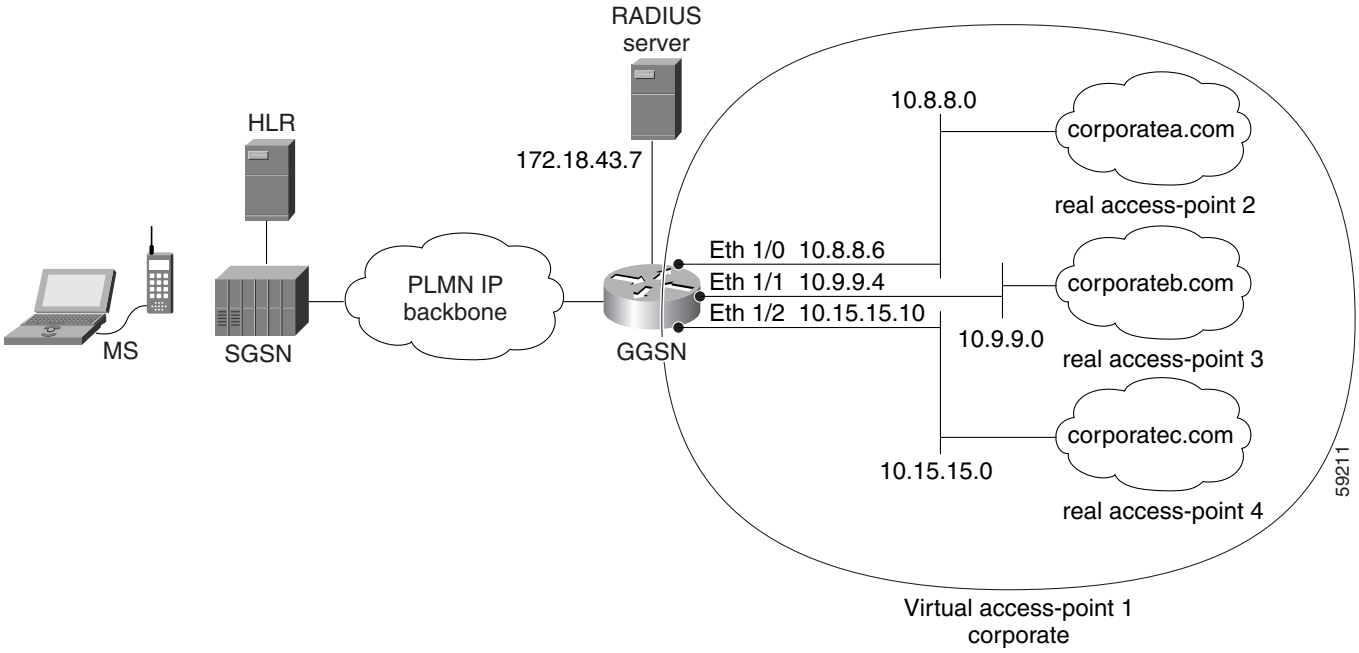
Virtual APN Configuration Example

The following example shows a GGSN that is configured for a virtual APN access point that serves as the focal connection for three different real corporate networks.

Notice the following areas in the GGSN configuration shown in this example:

- Three physical interfaces (Gi interfaces) are defined to establish access to the real corporate networks: Ethernet 1/0, Ethernet 1/1, and Ethernet 1/2.
- Four access points are configured:
  - Access point 1 is configured as the virtual access point with an APN called corporate. No other configuration options are applicable at the virtual access point. The "corporate" virtual APN is the APN that is provisioned at the HLR and DNS server.
  - Access points 2, 3, and 4 are configured to the real network domains: corporatea.com, corporateb.com, and corporatec.com. The real network domains are indicated in the PCO of the PDP context request.

Figure 6-2 Virtual APN Configuration Example



GGSN Configuration

```
!
version 12.2
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
! Enable the router for GGSN services
!
service gprs ggsn
!
hostname ggsn
```

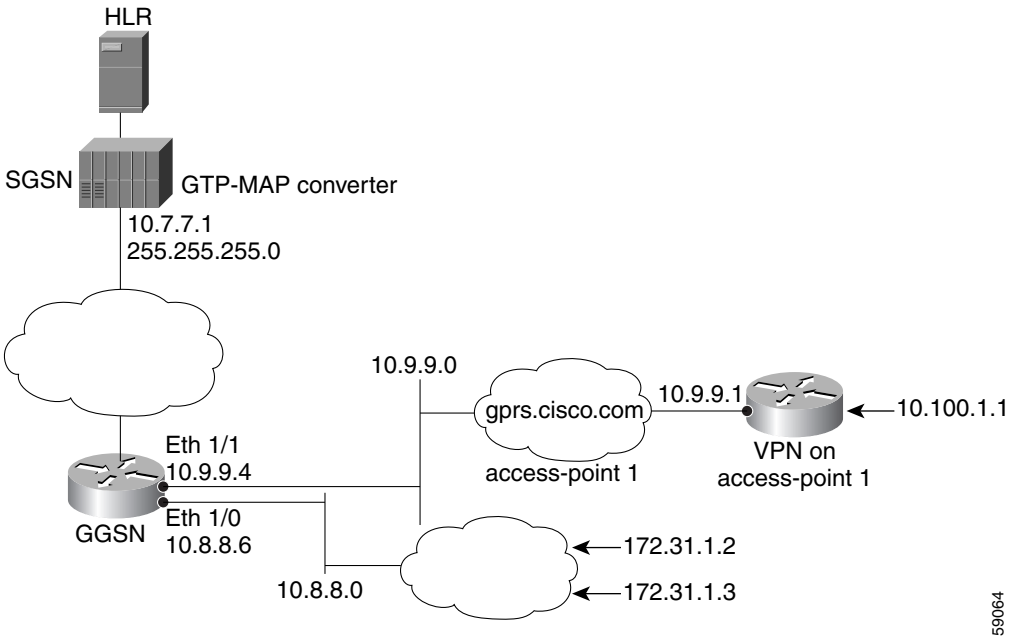
```
!
no logging buffered
logging rate-limit console 10 except errors
aaa new-model
aaa group server radius foo
 server 172.18.43.7 auth-port 1645 acct-port 1646
aaa authentication ppp foo group foo
aaa accounting network foo start-stop group foo

!
ip subnet-zero
!
!
no ip dhcp-client network-discovery
!
!
interface Loopback1
 ip address 10.2.3.4 255.255.255.255
!
interface FastEthernet0/0
 ip address 172.18.43.174 255.255.255.240
 duplex half
!
interface FastEthernet2/0
 description Gn interface
 ip address 192.168.10.56 255.255.255.0
!
! Define Gi physical interfaces to real networks
!
interface Ethernet1/0
 description Gi interface to corporatea.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 corporateb.com
 ip address 10.9.9.4 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex half
!
interface Ethernet1/2
 description Gi interface to corporatec.com
 ip address 10.15.15.10 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 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
!
ip default-gateway 172.18.43.161
ip kerberos source-interface any
ip classless
ip route 10.7.7.0 255.255.255.0 10.8.8.2
ip route 10.21.21.0 255.255.255.0 Ethernet1/1
ip route 10.102.82.0 255.255.255.0 172.18.43.161
```

```
ip route 192.168.1.1 255.255.255.255 FastEthernet2/0
ip route 172.18.0.0 255.255.0.0 172.18.43.161
no ip http server
!
gprs access-point-list gprs
!
! Configure a virtual access point called corporate
!
 access-point 1
   access-point-name corporate
   access-type virtual
   exit
!
! Configure three real access points called corporatea.com,
! corporateb.com, and corporatec.com
!
 access-point 2
   access-point-name corporatea.com
   access-mode non-transparent
   aaa-group authentication foo
   exit
 access-point 3
   access-point-name corporateb.com
   access-mode transparent
   ip-address-pool dhcp-client
   dhcp-server 10.21.21.1
   exit
   !
 access-point 4
   access-point-name corporatec.com
   access-mode non-transparent
   aaa-group authentication foo
   exit
   !
 !
gprs maximum-pdp-context-allowed 90000
gprs gtp path-echo-interval 0
gprs default charging-gateway 10.15.15.1
!
radius-server host 172.18.43.7 auth-port 1645 acct-port 1646 non-standard
radius-server retransmit 3
radius-server key 7 12150415
call rsvp-sync
!
no mgcp timer receive-rtcp
!
mgcp profile default
!
!
gatekeeper
 shutdown
!
end
```

Network-Initiated PDP Request Configuration Example

The following example shows a GGSN that is configured to support network initiated PDP contexts at a VPN on access point 1 for statically configured IP addresses. This example also shows support of network-initiated PDP contexts for MSs with an IP address of 172.31.1.2 and 172.31.1.3, which have been statically configured on the GGSN through access point 2.

Figure 6-3 Network Initiated PDP Request Configuration Example



GGSN Configuration

```
!
version 12.2
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
! Enable the router for GGSN services
!
service gprs ggsn
!
hostname ggsn
!
no logging buffered
logging rate-limit console 10 except errors
!
aaa new-model
!
aaa group server radius foo
 server 172.18.43.7 auth-port 1645 acct-port 1646
aaa authentication ppp foo group foo
aaa accounting network foo start-stop group foo
!
! Configure a VRF routing table
! and define an identifier
```

```
!
ip vrf vpn1
 rd 100:1
!
ip subnet-zero
!
no ip dhcp-client network-discovery
!
!
! Enable CEF switching
!
ip cef
!
interface Loopback1
 ip address 10.2.3.4 255.255.255.255
!
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
!
! Configure VRF at the interface
!
interface Ethernet1/1
 description Gi interface to gprs.cisco.com
 ip address 10.9.9.4 255.255.255.0
 ip vrf forwarding vpn1
 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
!
ip default-gateway 172.18.43.161
ip kerberos source-interface any
ip classless
ip route 10.7.7.0 255.255.255.0 10.8.8.2
ip route 10.102.82.0 255.255.255.0 172.18.43.161
ip route 192.168.0.0 255.255.0.0 172.18.43.161
ip route 172.18.0.0 255.255.0.0 172.18.43.161
no ip http server
!
gprs access-point-list gprs
!
! Configure an access point for gprs.cisco.com
! and enable network initiated PDP context support
! for a VPN
!
```

```
 access-point 1
   access-point-name gprs.cisco.com
   aaa-group authentication foo
!
! Enable network initiated PDP context support
!
   network-request-activation
!
! Configure VRF at the access point
!
   vrf vpn1
   exit
!
! Configure an access point for gprt.cisco.com
! and enable network-initiated PDP context support
!
 access-point 2
   access-point-name gprt.cisco.com
   network-request-activation
   exit
   !
 access-point 3
   access-point-name gpru.cisco.com
   access-mode non-transparent
   aaa-group authentication foo
   exit
   !
 !
gprs maximum-pdp-context-allowed 90000
gprs gtp path-echo-interval 0
gprs default charging-gateway 10.15.15.1
gprs gtp ip udp ignore checksum
!
! Configure the IP address of the SGSN to perform GTP-to-MAP and
! MAP-to-GTP conversion between the HLR and GGSN
!
gprs default map-converting-gsn 10.7.7.1
!
! Configure a static IP address to IMSI mapping for each MS
!
gprs ni-pdp ip-imsi single 1 10.100.1.1 11111111111111F1
gprs ni-pdp ip-imsi single 2 172.31.1.2 11111111111111F2
gprs ni-pdp ip-imsi single 2 172.31.1.3 11111111111111F3
!
!
radius-server host 172.18.43.7 auth-port 1645 acct-port 1646 non-standard
radius-server retransmit 3
radius-server key 7 12150415
call rsvp-sync
!
no mgcp timer receive-rtcp
!
mgcp profile default
!
!
gatekeeper
 shutdown
!
end
```

Blocking Access by Foreign Mobile Stations Configuration Example

The following example shows a partial configuration where access point 100 blocks access by foreign mobile stations:

```
!
version 12.2
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
! Enables the router for GGSN services
!
service gprs ggsn
!
hostname ggsn
!
gprs access-point-list gprs
!
access-point 100
 access-point-name blocking
!
! Enables blocking of MS to APN 100
! that are outside ! of the PLMN
!
 block-foreign-ms
exit
!
. . .
!
! Configures the MCC and MNC codes
!
gprs mcc 123 mnc 456
```

Duplicate IP Address Protection Configuration Example

The following example shows a partial configuration that specifies three different sets of IP address ranges used by the GPRS/UMTS network (which are thereby excluded from the MS IP address range):

```
gprs ms-address exclude-range 10.0.0.1 10.20.40.50
gprs ms-address exclude-range 172.16.150.200 172.30.200.255
gprs ms-address exclude-range 192.168.100.100 192.168.200.255
```

Configuring PPP Support on the GGSN

The GGSN supports the GTP with the Point to Point Protocol (PPP) in three different ways. The different types of PPP support on the GGSN are differentiated by where the PPP endpoints occur within the network, whether Layer 2 Tunneling Protocol (L2TP) is in use, and where IP packet service occurs. This chapter describes the different methods of PPP support on the GGSN and how to configure those methods.

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

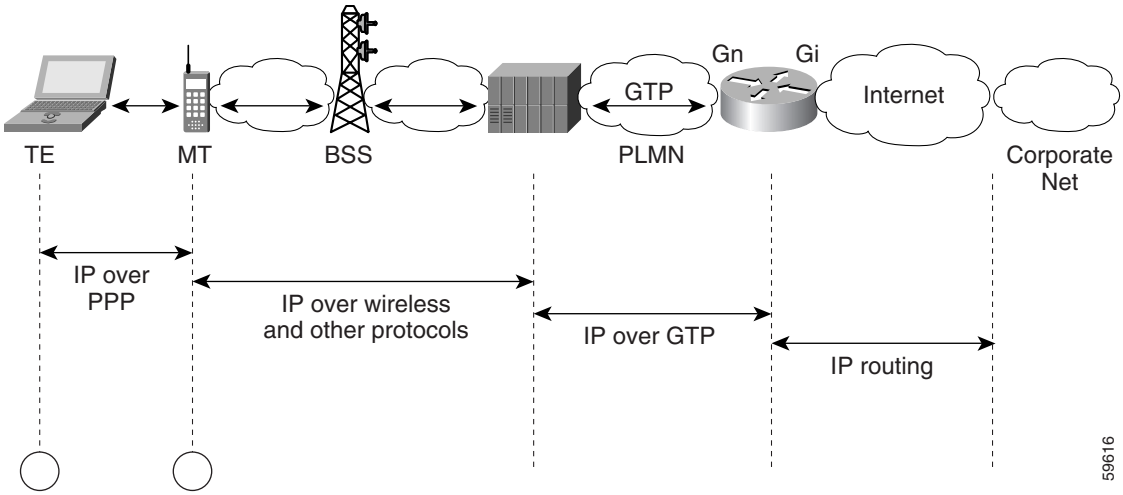
This chapter includes the following sections:

Overview of PPP Support on the GGSN

Before GGSN Release 3.0, the GGSN supported a topology of IP over PPP between the terminal equipment (TE) and mobile termination (MT). Only IP packet services and routing were supported from the MT through the SGSN, over the Gn interface and the GTP tunnel to the GGSN, and over the Gi interface to the corporate network. No PPP traffic flow was supported over the GTP tunnel, or between the GGSN and the corporate network.

Figure 7-1 shows the implementation of IP over GTP without any PPP support within a GPRS network.

Figure 7-1 IP Over GTP Topology Without PPP Support on the GGSN



The PPP PDP type was added to the GSM standards in GSM 04.08 version 7.4.0 and GSM 09.60 version 7.0.0. PPP is a widespread Layer 2 protocol that is frequently used in a variety of WAN environments, including frame relay, ATM, and X.25 networks.

PPP provides security checking through the Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP), and it uses the IP Control Protocol (IPCP) sublayer to negotiate IP addresses. Perhaps the most important characteristic of PPP support within the GPRS/UMTS network is PPP's tunneling capability through a virtual private data network (VPDN) using L2TP. Tunneling allows PPP sessions to be transported through public networks to a private corporate network, without any security exposure in between. Authentication and dynamic IP address allocation can be performed at the edge of the corporate network.

GGSN Release 3.0 and later provides the following three methods of PPP support on the GGSN:

- GTP-PPP
- GTP-PPP-L2TP
- GTP-PPP-Regeneration

Note Under optimal conditions, the GGSN supports 8000 PDP contexts when a PPP method is configured. However, the router platform, amount of memory installed, method of PPP support configured, and the rate of PDP context creation configured, will affect this number.

The following sections in this chapter describe each method in more detail, and describe how to configure and verify that type of PPP support on the GGSN.

Configuring GTP-PPP Termination on the GGSN

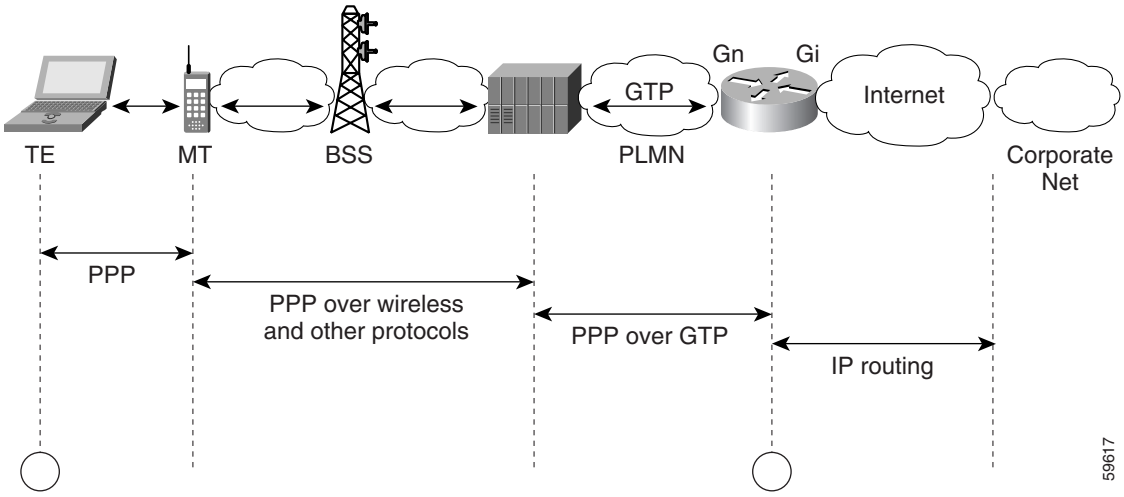
This section provides an overview of and describes how to configure PPP over GTP on the GGSN. It includes the following topics:

Overview of GTP-PPP Termination on the GGSN

The GGSN supports the PPP PDP type over GTP without using L2TP. In this topology, the GGSN provides PPP support from the terminal equipment (TE) and mobile termination (MT) or mobile station (MS) through the SGSN, over the Gn interface and the GTP tunnel to the GGSN. The PPP endpoints are at the terminal equipment (TE) and the GGSN. IP routing occurs from the GGSN over the Gi interface to the corporate network.

Figure 7-2 shows the implementation of PPP over GTP without L2TP support within a GPRS network.

Figure 7-2 PPP Over GTP Topology With PPP Termination the GGSN



Benefits

PPP over GTP support on the GGSN provides the following benefits:

- Different traffic types can be supported over GTP.
- Authentic negotiation of PPP options can occur for PPP endpoints (no need for proxy PPP negotiation).
- Provides the foundation for GTP to interwork with other PPP networking protocols, such as L2TP.

- Requirements for MT intelligence are simplified, with no need for support of a PPP stack on the MT.
- Additional session security is provided.
- Provides increased flexibility of IP address assignment to the TE.

Preparing to Configure PPP Over GTP on the GGSN

Before you begin to configure PPP over GTP support on the GGSN, you need to determine the method that the GGSN will use to allocate IP addresses to users. There are certain configuration dependencies based on the method of IP address allocation that you want to support.

Be sure that the following configuration guidelines are met to support the type of IP address allocation in use on your network:

- RADIUS IP address allocation
  - Be sure that users are configured on the RADIUS server using the complete username@domain format.
  - Specify the **no peer default ip address** command at the PPP virtual template interface.
  - For more information about configuring RADIUS services on the GGSN, see the "Configuring Security on the GGSN" chapter in this book.
- DHCP IP address allocation
  - Be sure that you configure the scope of the addresses to be allocated on the same subnet as the loopback interface.
  - Do not configure an IP address for users on the RADIUS server.
  - Specify the **peer default ip address dhcp** command at the PPP virtual template interface.
  - Specify the **aaa authorization network** *method_list* **none** command on the GGSN.
  - For more information about configuring DHCP services on the GGSN, see the "Configuring DHCP on the GGSN" chapter in this book.
- Local pool IP address allocation
  - Be sure to configure a local pool using the **ip local pool** command.
  - Be sure that you do not configure an IP address for users on the RADIUS server.
  - Specify the **peer default ip address pool** *pool-name* command.

GTP-PPP Termination Configuration Task List

To configure PPP over GTP support on the GGSN, perform the following tasks:

- Configuring a Loopback Interface, page 7-5 (Recommended)
- Configuring a PPP Virtual Template Interface, page 7-5 (Required)
- Associating the Virtual Template Interface for PPP on the GGSN, page 7-7 (Required)

Configuring a Loopback Interface

Cisco Systems recommends that you configure the virtual template interface as unnumbered, and associate its IP numbering with a loopback interface.

A loopback interface is a software-only interface that emulates an interface that is always up. It is a virtual interface supported on all platforms. The interface-number is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces you can create. The GGSN uses loopback interfaces to support the configuration of several different features.

To configure a loopback interface on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface loopback interface-number`	Defines a loopback interface on the GGSN, where *interface-number* identifies the loopback interface.
Step 2	`Router(config-if)# ip address ip-address mask [secondary]`	Specifies an IP address for the interface, where: • *ip-address*—Specifies the IP address of the interface in dotted decimal format. • *mask*—Specifies a subnet mask in dotted decimal format. • **secondary**—Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.

Configuring a PPP Virtual Template Interface

To support PPP over GTP, you must configure a virtual template interface on the GGSN that supports PPP encapsulation. Therefore, the GGSN will have two virtual template interfaces: one for GTP encapsulation and one for PPP encapsulation. The GGSN uses the PPP virtual template interface to create all PPP virtual access interfaces for PPP sessions on the GGSN.

Cisco Systems recommends that you configure the virtual template interface as unnumbered, and associate its IP numbering with a loopback interface.

Because it is the default, PPP encapsulation does not appear in the **show running-config** output for the interface.

To configure a PPP virtual template interface on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template number`	Creates a virtual template interface, where *number* identifies the virtual template interface. This command enters you into interface configuration mode. **Note** This number must match the *number* configured in the corresponding **gprs gtp ppp vtemplate** command.
Step 2	`Router(config-if)# ip unnumbered type number`	Enables IP processing on the virtual template interface without assigning an explicit IP address to the interface, where *type* and *number* specifies another interface for which the router has been assigned an IP address. For the GGSN, this can be a Gi interface or a loopback interface. Cisco Systems recommends using a loopback interface.
Step 3	`Router(config-if)# no peer default ip address` (for RADIUS server) or `Router(config-if)# peer default ip address dhcp` (for DHCP server) or `Router(config-if)# peer default ip address pool pool-name` (for local pool)	Specifies the prior peer IP address pooling configuration for the interface. If you are using a RADIUS server for IP address allocation, then you need to disable peer IP address pooling.
Step 4	`Router(config-if)# encapsulation ppp`	(Optional) Specifies PPP as the encapsulation type for packets transmitted over the virtual template interface. PPP is the default encapsulation. **Note** PPP is the default encapsulation and does not appear in the output of the **show running-config** command for the virtual template interface unless you manually configure the command.
Step 5	`Router(config-if)# ppp authentication {pap [chap]} [default]`	Enables CHAP or PAP or both and specifies the order in which CHAP and PAP authentication are selected on the interface, where • **pap** [**chap**]—Enables PAP, CHAP, or both on the interface. • **default**—Name of the method list created with the **aaa authentication ppp** command.

Associating the Virtual Template Interface for PPP on the GGSN

Before you associate the virtual template interface for PPP, you must configure the virtual template interface. The number that you configure for the virtual template interface must correspond to the number that you specify in the **gprs gtp ppp vtemplate** command.

To associate the virtual template interface for GGSN, use the following command in global configuration mode:

Command	Purpose
`Router(config)# gprs gtp ppp vtemplate number`	Associates the virtual template interface that defines the PPP characteristics with support for the PPP PDP type over GTP on the GGSN. **Note** This number must match the *number* configured in the corresponding **interface virtual-template** command.

Configuring GTP-PPP With L2TP on the GGSN

This section provides an overview of and describes how to configure PPP over GTP with L2TP support on the GGSN. It includes the following topics:

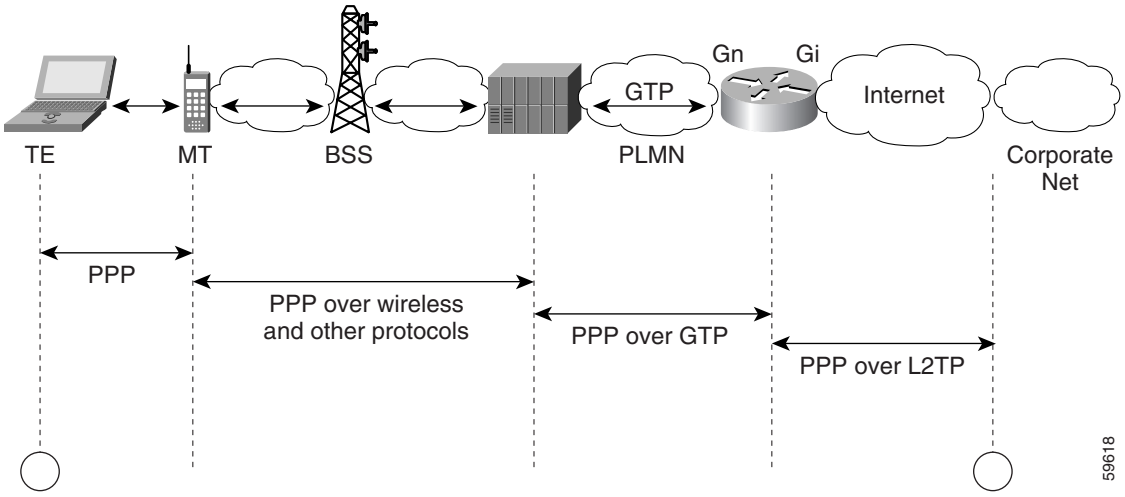
Overview of GTP-PPP With L2TP on the GGSN

The GGSN supports PPP over GTP using L2TP, without IP routing. The GGSN provides PPP support from the TE and MT through the SGSN, over the Gn interface and the GTP tunnel to the GGSN, and over the Gi interface and an L2TP tunnel to the corporate network. In this scenario, the PPP termination endpoints are at the TE and the L2TP network server (LNS) at the corporate network.

With L2TP support, packets are delivered to the LNS by routing L2TP- and PPP-encapsulated IP payload. Without L2TP, pure IP payload is routed to the LNS at the corporate network.

Figure 7-3 shows the implementation of PPP over GTP with L2TP support within a GPRS network.

Figure 7-3 PPP Over GTP With L2TP Topology on the GGSN



Benefits

PPP over GTP with L2TP support on the GGSN provides the following benefits:

- VPN security using L2TP tunnels provides secure delivery of user data over the public network to a corporate network.
- Real end-to-end PPP sessions, with authentication and address negotiation and assignment.
- Corporate networks can retain control over access to their servers and do not need to provide access by the GGSN to those servers.
- Configuration changes on corporate servers can occur without requiring an update to the GGSN.

GTP-PPP With L2TP Configuration Task List

Configuring GTP over PPP with L2TP requires many of the same configuration tasks as those required to configure GTP over PPP without L2TP, with some additional tasks to configure the GGSN as an L2TP access concentrator (LAC) and to configure AAA services.

To configure PPP over GTP with L2TP support on the GGSN, perform the following tasks:

- Configuring the GGSN as a LAC, page 7-9 (Required)
- Configuring AAA Services for L2TP Support, page 7-10 (Required)
- Configuring a Loopback Interface, page 7-12 (Recommended)
- Configuring a PPP Virtual Template Interface, page 7-12 (Required)
- Associating the Virtual Template Interface for PPP on the GGSN, page 7-13 (Required)

Configuring the GGSN as a LAC

When you use L2TP services on the GGSN to the LNS in the corporate network, you need to configure the GGSN as a LAC by enabling VPDN services on the GGSN.

For more information about VPDN configuration and commands in the Cisco IOS software, refer to the *Cisco IOS Dial Technologies Configuration Guide* and *Command Reference* publications.

To configure the GGSN as a LAC where the tunnel parameters are configured locally on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn enable`	Enables VPDN on the router and directs the router to look for tunnel definitions in a local database and on a remote authorization server (home gateway), if one is present. **Note** Only this step is required if you are using a RADIUS server to provide tunnel parameters.
Step 2	`Router(config)# vpdn-group group-number`	Defines a VPDN group and enters VPDN group configuration mode.
Step 3	`Router(config-vpdn)# request-dialin`	Enables the router to request dial-in tunnels and enters request dial-in VPDN sub-group configuration mode.
Step 4	`Router(config-vpdn-req-in)# protocol l2tp`	Specifies the L2TP protocol for dial-in tunnels.
Step 5	`Router(config-vpdn-req-in)# domain domain-name`	Specifies that users with this domain name will be tunnelled. Configure this command for every domain name you want to tunnel.
Step 6	`Router(config-vpdn-req-in)# exit`	Returns you to VPDN group configuration mode
Step 7	`Router(config-vpdn)# initiate-to ip ip-address [limit limit-number] [priority priority-number]`	Specifies the destination IP address for the tunnel.
Step 8	`Router(config-vpdn)# local name name`	Specifies the local name that is used to authenticate the tunnel.

Note You can configure the L2TP tunnel parameters locally on the GGSN, or the tunnel parameters can be provided by a RADIUS server. If a RADIUS server is providing the tunnel parameters, then in this procedure you only need to configure the **vpdn enable** command on the GGSN.

Configuring AAA Services for L2TP Support

Before the VPDN stack on the GGSN opens an L2TP tunnel to an LNS, it tries to authorize the tunnel first. The GGSN consults its local database to perform this authorization. Therefore, you need to configure the appropriate AAA services for the GGSN to support L2TP tunnel authorization. Note that this is for authorization of the tunnel itself—not for user authorization.

This section describes only those commands required to implement authorization for L2TP support on the GGSN. It does not describe all of the tasks required to configure RADIUS and AAA support on the GGSN. For more information about enabling AAA services and configuring AAA server groups on the GGSN, see the "Configuring Security on the GGSN" chapter in this book.

Note To correctly implement authentication and authorization services on the GGSN for L2TP support, you must configure the same methods and server groups for both.

To configure authorization for L2TP support on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa authorization network default local`	(Optional) Specifies that the GGSN consults its local database, as defined by the **username** command, for tunnel authorization.
Step 2	`Router(config)# aaa authorization network {default | list-name} group group-name [group group-name...]`	Specifies one or more authentication, authorization, and accounting (AAA) authentication methods for use on interfaces running PPP where, • **network—**Runs authorization for all network-related service requests, including SLIP1, PPP2, PPP NCPs3, and ARA4. • **default—**Uses the listed authentication methods that follow this argument as the default list of methods when a user logs in. • *list-name*—Specifies the character string used to name the list of authentication methods tried when a user logs in. • **group** *group-name*—Uses a subset of RADIUS servers for authentication as defined by the **aaa group server radius** command. **Note** Be sure to use a method list and do not use the **aaa authorization network default group radius** form of the command. For L2TP support, the *group-name* must match the group that you specify in the **aaa authentication ppp** command.
Step 3	`Router(config)# username name password secret`	Specifies the password to be used in CHAP caller identification, where *name* is the name of the tunnel. **Note** A username in the form of *ciscouser*, *ciscouser@corporate1.com*, and *ciscouser@corporate2.com* are considered to be three different entries. Repeat this step to add a username entry for each remote system from which the local router or access server requires authentication.

Note You can configure the L2TP tunnel parameters locally on the GGSN, or the tunnel parameters can be provided by a RADIUS server. If a RADIUS server is providing the tunnel parameters, then in this procedure you only need to configure the **username** command on the GGSN.

Configuring a Loopback Interface

Cisco Systems recommends that you configure the virtual template interface as unnumbered, and associate its IP numbering with a loopback interface.

A loopback interface is a software-only interface that emulates an interface that is always up. It is a virtual interface supported on all platforms. The interface-number is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces you can create. The GGSN uses loopback interfaces to support the configuration of several different features.

To configure a loopback interface on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface loopback interface-number`	Defines a loopback interface on the GGSN, where *interface-number* identifies the loopback interface.
Step 2	`Router(config-if)# ip address ip-address mask [secondary]`	Specifies an IP address for the interface, where: • *ip-address*—Specifies the IP address of the interface in dotted decimal format. • *mask*—Specifies a subnet mask in dotted decimal format. • **secondary**—Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.

Note IP addresses on the loopback interface are only needed for PPP PDPs not using L2TP. Cisco Systems recommends using IP addresses for the case where PPP PDPs are destined to a domain that is not configured with L2TP.

Configuring a PPP Virtual Template Interface

To support PPP over GTP, you must configure a virtual template interface on the GGSN that supports PPP encapsulation. Therefore, the GGSN will have two virtual template interfaces: one for GTP encapsulation and one for PPP encapsulation. The GGSN uses the PPP virtual template interface to create all PPP virtual access interfaces for PPP sessions on the GGSN.

Note If you are planning to support both GTP-PPP and GTP-PPP-L2TP (PPP PDPs with and without L2TP support), then you must use the same virtual template interface for PPP.

Cisco Systems recommends that you configure the virtual template interface as unnumbered, and associate its IP numbering with a loopback interface.

PPP is the default encapsulation, so it does not need to be explicitly configured. Because it is the default, PPP encapsulation does not appear in the **show running-config** output for the interface.

To configure a PPP virtual template interface on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template number`	Creates a virtual template interface, where *number* identifies the virtual template interface. This command enters you into interface configuration mode. **Note** This number must match the *number* configured in the corresponding **gprs gtp ppp vtemplate** command.
Step 2	`Router(config-if)# ip unnumbered type number`	Enables IP processing on the virtual template interface without assigning an explicit IP address to the interface, where *type* and *number* specifies another interface for which the router has been assigned an IP address. For the GGSN, this can be a Gi interface or a loopback interface. Cisco Systems recommends using a loopback interface.
Step 3	`Router(config-if)# encapsulation ppp`	(Optional) Specifies PPP as the encapsulation type for packets transmitted over the virtual template interface. PPP is the default encapsulation. **Note** PPP is the default encapsulation and does not appear in the output of the **show running-config** command for the virtual template interface unless you manually configure the command.

Associating the Virtual Template Interface for PPP on the GGSN

Before you associate the virtual template interface for PPP, you must configure the virtual template interface. The number that you configure for the virtual template interface must correspond to the number that you specify in the **gprs gtp ppp vtemplate** command.

To associate the virtual template interface for GGSN, use the following command in global configuration mode:

Command	Purpose
`Router(config)# gprs gtp ppp vtemplate number`	Associates the virtual template interface that defines the PPP characteristics with support for the PPP PDP type over GTP on the GGSN. **Note** This number must match the *number* configured in the corresponding **interface virtual-template** command.

Configuring GTP-PPP Regeneration on the GGSN

This section provides an overview of and describes how to configure PPP over GTP with L2TP support on the GGSN. It includes the following topics:

- Overview of GTP-PPP Regeneration on the GGSN, page 7-14
- GTP-PPP Regeneration Configuration Task List, page 7-15

Overview of GTP-PPP Regeneration on the GGSN

The GGSN supports PPP in two different areas of the network, with two different sets of PPP endpoints, and IP over GTP in between. First, IP over PPP is in use between the TE and MT. From there, IP packet support occurs between the MT through the SGSN, over the Gn interface and the GTP tunnel to the GGSN. The GGSN initiates a new PPP session on the Gi interface over an L2TP tunnel to the corporate network. So, the second set of PPP endpoints occur between the GGSN and the LNS at the corporate network.

PPP regeneration on the GGSN supports the use of an IP PDP type in combination with PPP and L2TP. For each IP PDP context that the GGSN receives at an access point that is configured to support PPP regeneration, the GGSN regenerates a PPP session. The GGSN encapsulates any tunnel packet data units (TPDUs) in PPP and L2TP headers as data traffic and forwards them to the LNS.

PPP regeneration on the GGSN implements virtual routing and forwarding (VRF) to handle overlapping IP addresses. A VRF routing table is automatically enabled at each APN when you configure PPP regeneration at that APN.

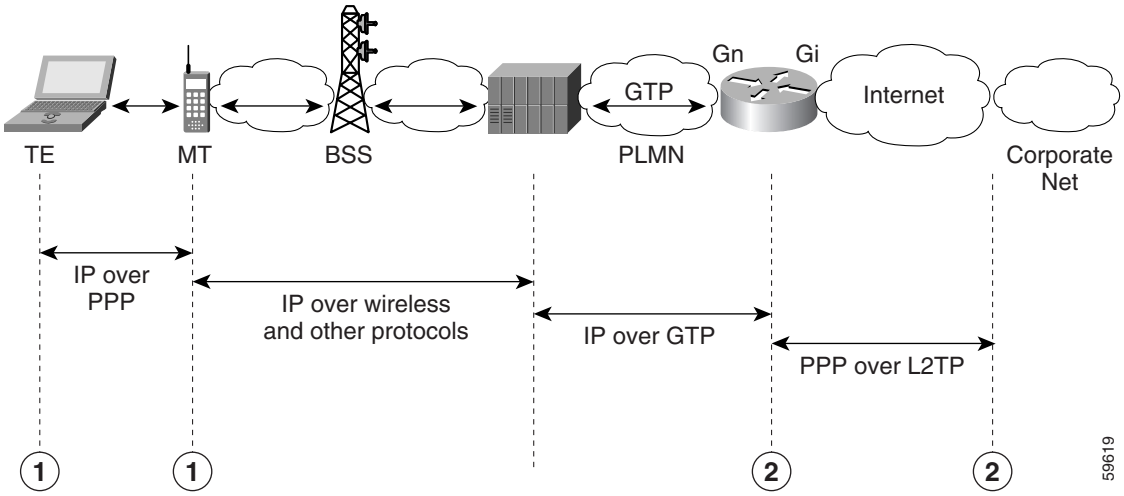
Restrictions

The GGSN supports PPP regeneration with the following restriction:

- Manual configuration of VRF is not supported.

Figure 7-4 shows the implementation of PPP support within a GPRS network using PPP regeneration on the GGSN.

Figure 7-4 PPP Regeneration Topology on the GGSN



GTP-PPP Regeneration Configuration Task List

Configuring IP over GTP with PPP regeneration on the GGSN requires similar configuration tasks as those required to configure GTP over PPP with L2TP, with some exceptions in the implementation.

To configure GTP-PPP regeneration support on the GGSN, perform the following tasks:

- Configuring the GGSN as a LAC, page 7-15 (Required)
- Configuring AAA Services for L2TP Support, page 7-16 (Required)
- Configuring a PPP Virtual Template Interface, page 7-18 (Required)
- Associating the Virtual Template Interface for PPP Regeneration on the GGSN, page 7-18 (Required)
- Configuring PPP Regeneration at an Access Point, page 7-19 (Required)

Configuring the GGSN as a LAC

When you use L2TP services on the GGSN to the LNS in the corporate network, you need to configure the GGSN as a LAC by enabling VPDN services on the GGSN.

For more information about VPDN configuration and commands in the Cisco IOS software, refer to the *Cisco IOS Dial Technologies Configuration Guide* and *Command Reference* publications.

To configure the GGSN as a LAC where the tunnel parameters are configured locally on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn enable`	Enables VPDN on the router and directs the router to look for tunnel definitions in a local database and on a remote authorization server (home gateway), if one is present. **Note** Only this step is required if you are using a RADIUS server to provide tunnel parameters.
Step 2	`Router(config)# vpdn-group group-number`	Defines a VPDN group and enters VPDN group configuration mode.
Step 3	`Router(config-vpdn)# request-dialin`	Enables the router to request dial-in tunnels and enters request dial-in VPDN sub-group configuration mode.
Step 4	`Router(config-vpdn-req-in)# protocol l2tp`	Specifies theL2TP protocol for dial-in tunnels.
Step 5	`Router(config-vpdn-req-in)# domain domain-name`	Specifies that users with this domain name will be tunnelled. Configure this command for every domain name you want to tunnel.
Step 6	`Router(config-vpdn-req-in)# exit`	Returns you to VPDN group configuration mode
Step 7	`Router(config-vpdn)# initiate-to ip ip-address [limit limit-number] [priority priority-number]`	Specifies the destination IP address for the tunnel.
Step 8	`Router(config-vpdn)# local name name`	Specifies the local name that is used to authenticate the tunnel.

Note You can configure the L2TP tunnel parameters locally on the GGSN, or the tunnel parameters can be provided by a RADIUS server. If a RADIUS server is providing the tunnel parameters, then in this procedure you only need to configure the **vpdn enable** command on the GGSN.

Configuring AAA Services for L2TP Support

Before the VPDN stack on the GGSN opens an L2TP tunnel to an LNS, it tries to authorize the tunnel first. The GGSN consults its local database to perform this authorization. Therefore, you need to configure the appropriate AAA services for the GGSN to support L2TP tunnel authorization. Note that this is for authorization of the tunnel itself—not for user authorization.

This section describes only those commands required to implement authorization for L2TP support on the GGSN. It does not describe all of the tasks required to configure RADIUS and AAA support on the GGSN. For more information about enabling AAA services and configuring AAA server groups on the GGSN, see the “Configuring Security on the GGSN” chapter in this book.

Note To correctly implement authentication and authorization services on the GGSN for L2TP support, you must configure the same methods and server groups for both.

To configure authorization for L2TP support on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa authorization network default local`	(Optional) Specifies that the GGSN consults its local database, as defined by the **username** command, for tunnel authorization.
Step 2	`Router(config)# aaa authorization network {default | list-name} group group-name [group group-name...]`	Specifies one or more authentication, authorization, and accounting (AAA) authentication methods for use on interfaces running PPP where, • **network—**Runs authorization for all network-related service requests, including SLIP1, PPP2, PPP NCPs3, and ARA4. • **default—**Uses the listed authentication methods that follow this argument as the default list of methods when a user logs in. • *list-name*—Specifies the character string used to name the list of authentication methods tried when a user logs in. • **group** *group-name*—Uses a subset of RADIUS servers for authentication as defined by the **aaa group server radius** command. **Note** Be sure to use a method list and do not use the **aaa authorization network default group radius** form of the command. For L2TP support, the *group-name* must match the group that you specify in the **aaa authentication ppp** command.
Step 3	`Router(config)# username name password secret`	Specifies the password to be used in CHAP caller identification, where *name* is the name of the tunnel. **Note** A username in the form of *ciscouser*, *ciscouser@corporate1.com*, and *ciscouser@corporate2.com* are considered to be three different entries. Repeat this step to add a username entry for each remote system from which the local router or access server requires authentication.

Note You can configure the L2TP tunnel parameters locally on the GGSN, or the tunnel parameters can be provided by a RADIUS server. If a RADIUS server is providing the tunnel parameters, then in this procedure you only need to configure the **username** command on the GGSN.

Configuring a PPP Virtual Template Interface

To support IP over GTP with PPP regeneration, you must configure a virtual template interface on the GGSN that supports PPP encapsulation. Therefore, the GGSN will have two virtual template interfaces: one for GTP encapsulation and one for PPP encapsulation. The GGSN uses the PPP virtual template interface to create all PPP virtual access interfaces for PPP sessions on the GGSN.

PPP is the default encapsulation, so it does not need to be explicitly configured. Because it is the default, PPP encapsulation does not appear in the **show running-config** output for the interface.

Be aware that the configuration commands for the PPP virtual template interface to support PPP regeneration on the GGSN is different from the previous configurations shown for GTP over PPP support.

To configure a PPP virtual template interface on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template number`	Creates a virtual template interface, where *number* identifies the virtual template interface. This command enters you into interface configuration mode. **Note** This number must match the *number* configured in the corresponding **gprs gtp ppp-regeneration vtemplate** command.
Step 2	`Router(config-if)# ip address negotiated`	Specifies that the IP address for a particular interface is obtained via PPP/IPCP (IP Control Protocol) address negotiation.
Step 3	`Router(config-if)# no peer neighbor-route`	Disables creation of neighbor routes.
Step 4	`Router(config-if)# encapsulation ppp`	(Optional) Specifies PPP as the encapsulation type for packets transmitted over the virtual template interface. PPP is the default encapsulation. **Note** PPP is the default encapsulation and does not appear in the output of the **show running-config** command for the virtual template interface unless you manually configure the command.
Step 5	`Router(config-if)# no ppp authentication`	(Optional) Disables PPP authentication. This is the default and does not appear in the output of the **show running-config** command.

Associating the Virtual Template Interface for PPP Regeneration on the GGSN

Before you associate the virtual template interface for PPP regeneration, you must configure a virtual template interface. The number that you configure for the virtual template interface must correspond to the number that you specify in the **gprs gtp ppp-regeneration vtemplate** command.

To associate the virtual template interface for PPP regeneration, use the following command in global configuration mode:

Command	Purpose
`Router(config)# gprs gtp ppp-regeneration vtemplate number`	Associates the virtual template interface that defines the PPP characteristics with support for the PPP regeneration on the GGSN. **Note** This number must match the *number* configured in the corresponding **interface virtual-template** command.

Configuring PPP Regeneration at an Access Point

To enable PPP regeneration on the GGSN, you must configure each access point for which you want to support PPP regeneration. There is not a global configuration command to enable PPP regeneration for all access points on the GGSN.

To create an access point and specify its type, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gprs access-point-list list-name`	Specifies a name for a new access point list, or references the name of the existing access point list and enters access-point list configuration mode.
Step 2	`Router(config-ap-list)# access-point access-point-index`	Specifies an index number for a new access point definition, or references an existing access point definition, and enters access point configuration mode.
Step 3	`Router(config-access-point)# access-point-name apn-name`	Specifies the access point network ID, which is commonly an Internet domain name. **Note** The *apn-name* must match the APN that has been provisioned at the MS, HLR, and DNS server.

	Command	Purpose
Step 4	`Router(config-access-point)# access-mode transparent`	(Optional) Specifies that no security authorization or authentication is requested by the GGSN for this access point. **Note** Transparent access is the default value, but it must be *manually* configured to support PPP regeneration at the access point if the access mode was previously non-transparent.
Step 5	`Router(config-access-point)# ppp-regeneration [max-session number] [setup-time seconds]`	Enables an access point to support PPP regeneration, where • **max-session** *number*—Specifies the maximum number of PPP regenerated sessions allowed at the access point. The default value is 65535. • **setup-time** *seconds*—Specifies the maximum amount of time (between 1 and 65535 seconds) within which a PPP regenerated session must be established. The default value is 60 seconds.

Monitoring and Maintaining PPP on the GGSN

This section provides a summary list of the **show** commands that you can use to monitor the different aspects of PPP configuration on the GGSN. Not all of the show commands apply to every method of configuration.

Use the following privileged EXEC commands to monitor and maintain PPP status on the GGSN:

Command	Purpose
`Router# show derived-config interface virtual-access number`	Displays the PPP options that GTP has configured on the virtual access interface for PPP regenerated sessions.
`Router# show gprs gtp pdp-context all`	Displays all currently active PDP contexts.
`Router# show gprs gtp pdp-context path ip-address`	Displays all currently active PDP contexts for the specified SGSN path.
`Router# show gprs gtp pdp-context pdp-type ppp`	Displays all currently active PDP contexts that are transmitted using PPP.
`Router# show gprs gtp status`	Displays information about the current status of the GTP on the GGSN.
`Router# show interfaces virtual-access number [configuration]`	Displays status, traffic data, and configuration information about a specified virtual access interface.
`Router# show vpdn session [all | packets | sequence | state | timers | window] [interface | tunnel | username]`	Displays VPN session information including interface, tunnel, username, packets, status, and window statistics.
`Router# show vpdn tunnel [all | packets | state | summary | transport] [id | local-name | remote-name]`	Displays VPN tunnel information including tunnel protocol, ID, local and remote tunnel names, packets sent and received, tunnel, and transport status.

Configuration Examples

This section provides configuration examples for the different types of PPP support on the GGSN. It includes the following examples:

- GTP-PPP Termination on the GGSN Configuration Example, page 7-21
- GTP-PPP Over L2TP Configuration Example, page 7-23
- GTP-PPP Regeneration Configuration Example, page 7-24
- AAA Services for L2TP Configuration Example, page 7-24

GTP-PPP Termination on the GGSN Configuration Example

The following example shows a GGSN configuration for GTP over PPP using PAP authentication using a RADIUS server at 172.16.0.2 to allocate IP addresses:

```
Router# show running-config
Building configuration...
Current configuration : 3521 bytes
!
version 12.2
no service single-slot-reload-enable
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
! Enables the router for GGSN services
!
service gprs ggsn
!
no logging buffered
logging rate-limit console 10 except errors
!
! Enables AAA globally
!
aaa new-model
!
! Defines AAA server group
!
aaa group server radius gtp_ppp
 server 172.16.0.2 auth-port 2001 acct-port 2002
!
! Configures authentication and authorization
! methods for PPP support.
!
aaa authentication ppp gtp_ppp group gtp_ppp
aaa authorization network gtp_ppp group gtp_ppp
aaa accounting network default start-stop group gtp_ppp
!
ip subnet-zero
!
! Configures a loopback interface
! for the PPP virtual template interface
!
interface Loopback2
 ip address 10.88.0.4 255.255.0.0
!
interface FastEthernet0/0
```

```
 description GN interface
 ip address 10.6.6.78 255.0.0.0
 no ip route-cache
 no ip mroute-cache
 duplex half
!
interface Ethernet2/0
 ip address 172.16.0.54 255.255.0.0
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet2/7
 ip address 10.7.0.1 255.255.0.0
 no ip route-cache
 no ip mroute-cache
!
interface FastEthernet3/0
 description Gi interface
 ip address 10.4.0.78 255.255.0.0
 no ip route-cache
 no ip mroute-cache
 duplex half
!
! Configures a VT interface for
! GTP encapsulation
!
interface loopback 1
 ip address 10.30.30.1 255.255.255.0
!
interface Virtual-Template1
 ip unnumber loopback 1
 encapsulation gtp
 gprs access-point-list gprs
!
! Configures a VT interface for
! PPP encapsulation
!
interface Virtual-Template2
 ip unnumbered Loopback2
 no ip route-cache
 no peer default ip address
 ppp authentication pap
!
ip kerberos source-interface any
ip classless
ip route 172.16.0.0 255.255.0.0 Ethernet2/0
no ip http server
!
gprs access-point-list gprs
  access-point 1
   access-point-name gprs.cisco.com
   aaa-group authentication gtp_ppp
   aaa-group accounting gtp_ppp
   exit
   !
! Associates the PPP virtual template
! interface for use by the GGSN
!
gprs gtp ppp-vtemplate 2
gprs default charging-gateway 10.7.0.2
!
! Configures a global RADIUS server host
! and specifies destination ports for
! authentication and accounting requests
```

```
!
radius-server host 172.16.0.2 auth-port 2001 acct-port 2002
radius-server retransmit 3
radius-server key cisco
!
!
end
```

GTP-PPP Over L2TP Configuration Example

The following example shows a partial configuration of the GGSN to support PPP over GTP with L2TP. Tunnel parameters are configured locally on the GGSN and are not provided by a RADIUS server:

```
. . .
!
! Enables AAA globally
!
aaa new-model
!
aaa authorization network default local
!
vpdn enable
!
! Configures a VPDN group
!
vpdn-group 1
  request-dialin
  protocol l2tp
  domain ppp-lns
  initiate-to ip 4.0.0.78 priority 1
  local name nas
!
! Configures a loopback interface
! for the PPP virtual template interface
!
interface Loopback2
 ip address 10.88.0.1 255.255.255.255
!
interface Virtual-Template2
 description VT for PPP L2TP
 ip unnumbered Loopback2
 no peer default ip address
 no peer neighbor-route
 ppp authentication pap chap
!
gprs access-point-list gprs
 access-point 15
 access-point-name ppp-lns
 exit
!
! Associates the PPP virtual template
! interface for use by the GGSN
!
gprs gtp ppp vtemplate 2
!
. . .
!
```

GTP-PPP Regeneration Configuration Example

The following example shows a partial configuration of the GGSN to support IP over GTP with PPP regeneration on the GGSN. Tunnel parameters are configured locally on the GGSN and are not provided by a RADIUS server:

```
!
. . .
!
! Enables AAA globally
!
vpdn enable
!
! Configures a VPDN group
!
vpdn-group 1
 request-dialin
  protocol l2tp
  domain ppp_regen1
 initiate-to ip 4.0.0.78 priority 1
 l2tp tunnel password 7 0114161648
!
! Configures a virtual template
! interface for PPP regeneration
!
interface Virtual-Template2
 description VT for PPP Regen
 ip address negotiated
 no peer neighbor-route
!
gprs access-point-list gprs
 access-point 6
  access-point-name ppp_regen1
  ppp-regeneration
  exit
!
! Associates the PPP-regeneration
! virtual template interface for use by the GGSN
!
gprs gtp ppp-regeneration vtemplate 2
```

AAA Services for L2TP Configuration Example

L2TP support is used on the GGSN to support both the PPP over GTP topology and the IP over GTP with PPP regeneration topology. The following examples shows a partial configuration of RADIUS and AAA services on the GGSN to provide L2TP support:

```
!
! Enables AAA globally
!
aaa new-model
!
! Defines AAA server group
!
aaa group server radius gtp_ppp
 server 172.16.0.2 auth-port 2001 acct-port 2002
!
! Configures authentication and authorization
! method gtp_ppp and AAA server group gtp_ppp
! for PPP support.
```

```
!
! NOTE: You must configure the same methods and groups
! to support L2TP as shown by the
! aaa authentication ppp gtp_ppp
! and aaa authorization network gtp_ppp commands.
!
aaa authentication ppp gtp_ppp group gtp_ppp
aaa authorization network default local
aaa authorization network gtp_ppp group gtp_ppp
aaa accounting network default start-stop group radius
username nas password 0 lab
username hgw password 0 lab
!
. . .
!
! Configures a global RADIUS server host
! and specifies destination ports for
! authentication and accounting requests
!
radius-server host 172.16.0.2 auth-port 2001 acct-port 2002
radius-server retransmit 3
radius-server key cisco
!
. . .
!
```

Optimizing GGSN Performance

This chapter describes how to optimize performance on the GGSN.

For a complete description of the GGSN commands in this chapter, refer to the *Cisco IOS Mobile Wireless 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:

Configuring Switching Paths on the GGSN

GGSN Release 3.0 and later implements the enhanced performance support of Cisco Express Forwarding (CEF) switching on the Gn and Gi interfaces on the GGSN. Prior to GGSN Release 3.0, only fast switching was supported. GGSN Release 4.0 only supports process switching and CEF switching.

This section provides an overview of the available switching paths on the GGSN and describes how to configure CEF switching. It includes the following topics:

Overview of Switching Paths

Cisco Systems routers provide a variety of switching algorithms, or paths, to optimize IP packet processing. Switching paths minimize the amount of time it takes for the router to forward a packet from its incoming interface to an outgoing interface. Some of the switching paths supported are dependent upon the type of router platform in use.

The following are some of the types of switching paths that are available on the router platforms, in order of performance benefit:

- Process switching
- Fast switching
- CEF switching

Note GGSN Release 3.0 and later supports process and CEF switching.

Process Switching

When packets are process switched, frames received by the router are copied into a system buffer. The router looks up the destination Layer 3 network address in its routing table and initializes the route-cache table. Packets are placed in a processing queue according to their Layer 3 protocol, and remain in the queue until the scheduler gives the CPU to the corresponding process. The waiting time depends on the number of processes waiting to run and the number of packets waiting to be processed. The routing decision is made based on the routing table and the Address Resolution Protocol (ARP) cache. When the routing decision is made, the packet is forwarded to the corresponding outgoing interface.

Fast Switching

When packets are fast switched, frames received by the router are immediately processed. The processor looks up the destination Layer 3 network address in the route-cache table on the router. If the destination is found in the cache table, the router rewrites the header and forwards the packet to the appropriate outgoing interface. If the destination address is not found, the packet is process switched and a route-cache entry is added for the new destination.

CEF Switching

CEF switching uses a forwarding information base (FIB) table and an adjacency table to accomplish packet switching. The adjacency table is indexed by Layer 3 network addresses and contains the corresponding Layer 2 information to forward a packet.

CEF switching eliminates the use of the route-cache table, and the overhead that is required in aging out its table entries and repopulating the table. The FIB table mirrors the entire contents of the IP routing table, which eliminates the need for a route-cache table.

For more information about switching paths, refer to the *Cisco IOS Switching Services Configuration Guide*, Release 12.2.

CEF Switching Configuration Task List

CEF switching is the preferred switching path, and it is required to support virtual routing and forwarding (VRF) on the GGSN. To achieve the maximum performance benefits of CEF switching on the GGSN, you should configure all of the following entities for CEF switching:

- Virtual template interface of the GGSN
- Physical interfaces that support GTP on the SGSN (the Gn interface)
- Physical interfaces over which MSs will access the PDNs (the Gi interface)

To configure CEF switching for GPRS/UMTS, perform the following tasks:

- Enabling CEF Switching Globally, page 8-3 (Required)
- Enabling CEF Switching on a Physical Interface, page 8-3 (Optional)

Enabling CEF Switching Globally

When you enable CEF switching globally on the GGSN, all interfaces on the GGSN are automatically enabled for CEF switching.

Note To ensure that CEF switching functions properly, wait a short period of time before enabling CEF switching after it has been disabled using the **no ip cef** command.

To enable CEF switching on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip cef`	Enables CEF on the route processor card.
Step 2	`Router(config)# gprs gtp ip udp ignore checksum`	Disables verification of the UDP checksum to support CEF switching on the GGSN.

Caution If you do not configure the **gprs gtp ip udp ignore checksum** command, G-PDUs (GTP PDUs) with a non-zero User Datagram Protocol (UDP) checksum will be process switched.

Enabling CEF Switching on a Physical Interface

After you have enabled CEF switching globally on the GGSN, CEF switching is automatically enabled on all of the physical interfaces.

If the **no ip route-cache cef** command is configured on the Gn or Gi interfaces of the GGSN, then you should enable CEF switching on those interfaces.

Note When CEF switching is enabled on the physical interface (either by configuration, or automatically, through the use if the **ip cef** global configuration command), the **ip route-cache cef** command does not appear in the output of the **show running-config** command. However, the **no ip route-cache cef** command does appear if it is configured.

To enable CEF switching on the physical interface between the GGSN and SGSN (over the Gn interface), and between the GGSN and PDNs (over the Gi interface), use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type slot/port`	Accesses the physical interface configuration. **Note** The actual syntax of the **interface** command depends on the type of physical interface that you have configured for the Gn or Gi interfaces.
Step 2	`Router(config-if)# ip route-cache cef`	Reenables disabled CEF or dCEF operation on an interface.

Verifying the CEF Switching Configuration

To verify that CEF switching is enabled for the GGSN on the virtual-template interface and the physical interfaces, use the **show running-config** and the **show cef interface** commands.

Step 1 Use the **show running-config** command and verify that the **ip cef** and **gprs gtp ip udp ignore checksum** commands are configured. The following example provides portions of output from a **show running-config** command showing the related commands in bold:

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

Current configuration :3815 bytes
!
version 12.2

...

service gprs ggsn

...

ip cef
ip address-pool dhcp-proxy-client
!
interface Ethernet1/0
 description - Ga interface to Charging Gateway
 ip address 10.67.67.1 255.255.255.0
 no ip mroute-cache
!
interface FastEthernet2/0
 description - Gn Interface to SGSN
 ip address 10.12.12.1 255.255.255.0
 no ip mroute-cache
 duplex half
!
interface FastEthernet4/0
 description - Gi Interface to PDN
 ip address 10.78.78.1 255.255.255.0
 no ip mroute-cache
 duplex full
!
interface loopback 1
```

```
 ip address 10.112.112.1 255.255.255.0
!
interface Virtual-Template1
 ip unnumber loopback 1
 encapsulation gtp
 gprs access-point-list gprs
!

...

gprs maximum-pdp-context-allowed 45000
gprs qos map canonical-qos
gprs default ip-address-pool dhcp-proxy-client
gprs default charging-gateway 10.65.65.2
gprs gtp ip udp ignore checksum
!
```

Step 2 Use the **show cef interface** command to verify that CEF switching is enabled on the virtual template interface, as shown in bold in the following example:

```
Router# show cef interface virtual-access 1
Virtual-Access1 is up (if_number 17)
  Corresponding hwidb fast_if_number 17
  Corresponding hwidb firstsw->if_number 17
  Internet address is 112.112.112.1/24
  ICMP redirects are always sent
  Per packet load-sharing is disabled
  IP unicast RPF check is disabled
  Inbound access list is not set
  Outbound access list is not set
  IP policy routing is disabled
  Interface is marked as point to point interface
  Hardware idb is Virtual-Access1
  Fast switching type 22, interface type 21
  IP CEF switching enabled
  IP Feature Fast switching turbo vector
  IP Feature CEF switching turbo vector
  Input fast flags 0x20000, Output fast flags 0x20000
  ifindex 13(13)
  Slot -1 Slot unit 1 VC -1
  Transmit limit accumulator 0x0 (0x0)
  IP MTU 1500
```

Step 3 Use the **show cef interface** command for each of the Gn and Gi interfaces to verify that CEF switching is enabled on those interfaces, as shown in bold in the following example. The following example shows sample output for the Gn interface:

```
Router# show cef interface fa2/0
FastEthernet2/0 is up (if_number 12)
  Corresponding hwidb fast_if_number 12
  Corresponding hwidb firstsw->if_number 12
  Internet address is 12.12.12.1/24
  ICMP redirects are always sent
  Per packet load-sharing is disabled
  IP unicast RPF check is disabled
  Inbound access list is not set
  Outbound access list is not set
  IP policy routing is disabled
  Hardware idb is FastEthernet2/0
  Fast switching type 1, interface type 18
  IP CEF switching enabled
  IP Feature Fast switching turbo vector
  IP Feature CEF switching turbo vector
  Input fast flags 0x0, Output fast flags 0x0
```

```
ifindex 10(10)
Slot 2 Slot unit 0 VC -1
Transmit limit accumulator 0x0 (0x0)
IP MTU 1500
```

Monitoring and Maintaining CEF Switching

This section describes the commands used to display CEF switching status on the GGSN and to display CEF packet processing for PDP contexts.

It includes the following topics:

- Show Command Summary, page 8-6
- Displaying CEF Switching Information for a PDP Context, page 8-6

Show Command Summary

This section provides a summary list of the **show** commands that you can use to monitor CEF switching on the GGSN.

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

Command	Purpose
Router# **show cef interface**	Displays CEF-related interface information.
Router# **show gprs gtp pdp-context imsi** *hex-data*	Displays PDP contexts by International Mobile Subscriber Identity (IMSI).
Router# **show gprs gtp pdp-context tid** *hex-data*	Displays PDP contexts by tunnel ID.

Displaying CEF Switching Information for a PDP Context

To display CEF packet processing statistics for a particular PDP context, you can use the **show gprs gtp pdp-context** command using the **tid** or **imsi** keywords. The following example shows sample output for the **show gprs gtp pdp-context tid** command for a PDP context. The output fields displaying CEF packet information are shown in bold:

```
Router #show gprs gtp pdp-context tid 2424111111111120
TID              MS Addr         Source  SGSN Addr       APN
2424111111111120 10.24.24.28     Static  10.10.10.2      www.pdn1.com

    current time :Dec 06 2001 13:15:34
    user_name (IMSI): 424211111111110    MS address: 10.24.24.28
    MS International PSTN/ISDN Number (MSISDN): 243926901
    sgsn_addr_signal: 10.10.10.2         ggsn_addr_signal: 10.30.30.1
    signal_sequence:  7                  seq_tpdu_up:      0
    seq_tpdu_down:    5380
    upstream_signal_flow:   371          upstream_data_flow:   372
    downstream_signal_flow: 1            downstream_data_flow: 1
    RAupdate_flow:          0
    pdp_create_time:   Dec 06 2001 09:54:43
    last_access_time:  Dec 06 2001 13:15:21
    mnrgflag:          0                tos mask map: 00
```

```
gtp pdp idle time: 72
gprs qos_req: 091101                canonical Qos class(req.): 01
gprs qos_neg: 25131F                canonical Qos class(neg.): 01
effective bandwidth: 0.0
rcv_pkt_count:       10026          rcv_byte_count:  1824732
send_pkt_count:      5380           send_byte_count: 4207160
cef_up_pkt:          10026          cef_up_byte:     1824732
cef_down_pkt:        5380           cef_down_byte:   4207160
cef_drop:            0
charging_id:         12321224
pdp reference count: 1
ntwk_init_pdp:       0
```

Minimizing Static Routes on the GGSN Using Route Aggregation

As of Release 3.0, the GGSN supports route aggregation to minimize the resource and performance impact of static routes for each PDP context request on the GGSN.

This section provides an overview of route aggregation and how to configure it on the GGSN. It includes the following topics:

Overview of Route Aggregation on the GGSN

The GGSN uses a static host route to forward user data packets received from the Gi interface, using the virtual template interface of the GTP tunnel, to the Gn interface.

Without route aggregation, the GGSN creates a static host route for each MS PDP request. For example, for 90,000 PDP contexts supported, the GGSN creates 90,000 static host routes in its IP routing table. These routing table entries are in addition to entries in the forwarding information base (FIB) table or fast switching cache. As the number of PDP contexts supported by the GGSN increases, the forwarding performance can be degraded and memory usage is increased.

To minimize the allocation of static routes in the IP routing table on the GGSN, you can specify that the GGSN creates a single network route for PDP contexts coming from a particular IP network.

If you use DHCP and route aggregation on the GGSN, you can control the IP address ranges assigned to PDP context requests from an MS to a particular PDN, and then you also can control how the GGSN aggregates those routes.

Use care when assigning IP addresses to an MS before you configure the aggregation ranges on the GGSN. A basic guideline is to aggregate as many addresses as possible, but to minimize your use of aggregation with respect to the total amount of IP address space being used by the access point.

As with other access point configuration options on the GGSN, you can configure route aggregation globally for all access points, or for a particular access point. At an access point, you can specify one or more **aggregate** commands. Or, you can configure the GGSN to establish automatic route aggregation for IP address masks returned by a DHCP or RADIUS server through a particular access point.

The way that the GGSN implements aggregate and static routes depends upon whether the IP address of the MS is statically or dynamically derived, and also upon the type of global and access point configuration that is being supported on the GGSN.

The following scenarios describe how the GGSN manages routes for MSs through an access point, for the possible route aggregation configurations and addressing methods:

- No aggregation is configured on the GGSN, at the APN or globally—The GGSN inserts the 32-bit host route of the MS into its routing table as a static route.
- A default aggregate route is configured globally, but no aggregation is configured at the APN:
  - If a statically or dynamically derived address for an MS matches the default aggregate route range, the GGSN inserts an aggregate route into its routing table.
  - If the MS address does not match the default aggregate route, the GGSN inserts the 32-bit host route as a static route into the routing table.
- A default aggregate route is configured globally, and automatic route aggregation is configured at the APN:
  - If a statically derived address for an MS matches the default aggregate route range, the GGSN inserts an aggregate route into its routing table.
  - If a statically derived address for an MS does not match the default aggregate route, the GGSN inserts the 32-bit host route as a static route into its routing table.
  - If a dynamically derived address for an MS is received, the GGSN aggregates the route based on the address and mask returned by the DHCP or RADIUS server.
- A default aggregate route is configured globally, and an aggregate route is also configured at the APN:
  - If a statically or dynamically derived address for an MS matches the aggregate range at the APN through which it was processed, or otherwise matches the default aggregate range, the GGSN inserts an aggregate route into its routing table.
  - If a statically or dynamically derived address for an MS does not match either the aggregate range at the APN, or the global default aggregate range, the GGSN inserts the 32-bit host route as a static route into its routing table.

Route Aggregation Configuration Task List

This section describes how to configure route aggregation on the GGSN. You can configure route aggregation globally on the GGSN for all access points, and you can configure individual aggregate routes or automatic route aggregation at a particular access point.

To configure route aggregation on the GGSN, perform the following tasks:

- Configuring Route Aggregation Globally on the GGSN, page 8-9 (Optional)
- Configuring Route Aggregation at an Access Point, page 8-9 (Optional)
- Configuring Automatic Route Aggregation at an Access Point, page 8-10 (Optional)

Note The **aggregate** and **gprs default aggregate** commands affect routing on the GGSN. Use care when planning and configuring IP address aggregation.

Configuring Route Aggregation Globally on the GGSN

You can configure aggregate routes globally to reduce the number of static routes implemented for PDP requests at all access points on the GGSN. The GGSN allows you to specify an IP network prefix to combine the routes of PDP requests from the same network as a single route on the GGSN.

If you use the **gprs default aggregate** command to globally define an aggregate IP network address range for all access points on the GGSN, you also can use the **aggregate** command to override this default address range at a particular access point.

Automatic route aggregation can be configured at the access-point configuration level only on the GGSN. The **gprs default aggregate** global configuration command does not support the auto option; therefore, you cannot configure automatic route aggregation globally on the GGSN. For more information about configuring automatic route aggregation, see the "Configuring Automatic Route Aggregation at an Access Point" section on page 8-10.

To specify that the GGSN configures an aggregate route in its IP routing table for any PDP context request from MSs on the specified network for any GGSN access point, use the following global configuration command:

Command	Purpose
`Router(config)# gprs default aggregate ip-network-prefix {/mask-bit-length | ip-mask}`	Specifies a global aggregate route in the IP routing table of the GGSN to route PDP requests at all access points on the GGSN.

Configuring Route Aggregation at an Access Point

You can configure aggregate routes to reduce the number of static routes implemented by the GGSN for PDP contexts at a particular access point. The GGSN allows you to specify an IP network prefix to combine the routes of PDP contexts from the same network as a single route on the GGSN.

You can specify multiple aggregate commands at each access point to support multiple network aggregates. However, if you use the **aggregate auto** command at the APN, you cannot specify any other aggregate route ranges at the APN. In this case, if you also need to handle other static route cases at the APN, then you need to use the **gprs default aggregate** global configuration command. For more information about configuring route aggregation globally, see the "Configuring Route Aggregation Globally on the GGSN" section on page 8-9.

To configure aggregate routes for a particular access point, or to override the default aggregate route specified by the **gprs default aggregate** global configuration command, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gprs access-point-list list_name`	Specifies the access-point list name and enters access-point list configuration mode.

	Command	Purpose
Step 2	`Router(config-ap-list)# access-point access-point-index`	Specifies a number associated with this access-point definition and enters access point configuration mode. If the access point is already defined, specify the number of the access point that you want to modify.
Step 3	`Router(config-access-point)# aggregate {auto | ip-network-prefix{/mask-bit-length | ip-mask}}`	Configures the GGSN to create an aggregate route in its IP routing table when receiving PDP requests from MSs on the specified network through a particular access point on the GGSN.

Configuring Automatic Route Aggregation at an Access Point

If you want the GGSN to automatically aggregate routes from a DHCP or RADIUS server for a particular access point, you can use the **aggregate auto** command at the access point. Automatic route aggregation is typically used when you are using a DHCP server at the access point.

Automatic route aggregation is not available through a global configuration for all access points on the GGSN. Therefore, to use automatic route aggregation, you must configure it at each access point where it applies.

In addition, if you use the **aggregate auto** command at the APN, you cannot specify any other aggregate route ranges at the APN. If you need to handle other static route cases at the APN, then you will have to use the **gprs default aggregate** global configuration command. For more information about configuring route aggregation globally, see the "Configuring Route Aggregation Globally on the GGSN" section on page 8-9.

To configure automatic route aggregation at an access point using DHCP services, perform the following tasks:

- Configuring the Access Point for Automatic Route Aggregation Using a DHCP Server, page 8-10
- Configuring a Loopback Interface for the DHCP Gateway Address, page 8-11

Configuring the Access Point for Automatic Route Aggregation Using a DHCP Server

To configure the GGSN to automatically aggregate routes for IP address masks that are returned by a DHCP server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gprs access-point-list list_name`	Specifies the access-point list name and enters access-point list configuration mode.
Step 2	`Router(config-ap-list)# access-point access-point-index`	Specifies a number associated with this access-point definition and enters access point configuration mode. If the access point is already defined, specify the number of the access point that you want to modify.
Step 3	`Router(config-access-point)# aggregate auto`	Configures the GGSN to automatically create an aggregate route in its IP routing table according to the IP address masks it receives from a DHCP or RADIUS server through a particular access point.

	Command	Purpose
Step 4	`Router(config-access-point)# ip-address-pool dhcp-proxy-client`	Specifies a dynamic address allocation method using IP address pools for the current access point, where **dhcp-proxy-client** specifies that the access point IP address pool is maintained on a DHCP server.
Step 5	`Router(config-access-point)# dhcp-server {ip-address | name} [{ip-address | name}]`	Specifies a primary (and backup) DHCP server that the GGSN uses at a particular access point to obtain IP address leases for mobile users for access to a PDN, where: • *ip-address*—Specifies the IP address of a DHCP server. The second (optional) *ip-address* argument specifies the IP address of a backup DHCP server. • *name*—Specifies the host name of a DHCP server. The second (optional) *name* argument specifies the host name of a backup DHCP server.
Step 6	`Router(config-access-point)# dhcp-gateway-address ip-address`	Specifies the subnet in which the DHCP server should return addresses for DHCP requests for MS users entering a particular PDN access point.

Configuring a Loopback Interface for the DHCP Gateway Address

When you configure DHCP services at an APN and you specify a DHCP gateway address, you need to configure a loopback interface on the GGSN that corresponds to the IP address of the DHCP gateway.

To configure a loopback interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface loopback number`	Defines a loopback interface on the GGSN with a particular reference number.
Step 2	`Router(config-if)# ip address ip-address mask`	Specifies an IP address for the interface, where: • *ip-address*—Specifies the IP address of the interface in dotted decimal format. **Note** The *ip-address* should match the IP address in the **dhcp-gateway-address** command. • *mask*—Specifies a subnet mask in dotted decimal format.

Verifying Aggregate Routes on the GGSN

To verify the route aggregation configuration on the GGSN, use the **show running-config**, **show gprs gtp pdp-context all**, **show ip route**, and **show gprs-access point** commands.

Step 1 Use the **show running-config** command and verify that the **gprs default aggregate** global configuration or the **aggregate** access point configuration commands are configured.

The following example provides portions of output from a **show running-config** command showing areas of the configuration that are related to the aggregate route configuration in bold. Note that two **aggregate** commands are configured at access point 8, and a global **gprs default aggregate** route is also configured.

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

Current configuration :3815 bytes
!
version 12.2

...

service gprs ggsn

...

ip cef
ip address-pool dhcp-proxy-client
!
interface Loopback0
 ip address 10.88.0.1 255.255.255.255
!
. . .
!
interface Virtual-Template1
 ip unnumber Loopback0
 encapsulation gtp
 gprs access-point-list gprs
!
. . .
!
gprs access-point-list gprs
  access-point 8
   access-point-name pdn.aaaa.com
   aggregate 10.88.0.0 255.255.255.0
   aggregate 10.80.0.0 255.255.255.0
   exit

...

gprs maximum-pdp-context-allowed 45000
gprs qos map canonical-qos
gprs default ip-address-pool dhcp-proxy-client
gprs default charging-gateway 10.65.65.2
gprs default aggregate 192.168.100.0/24
gprs gtp ip udp ignore checksum
```

Step 2 Use the **show gprs gtp pdp-context all** command to verify the PDP context requests that are active on the GGSN.

In the following output for the **show gprs gtp pdp-context all** command, 5 PDP context requests are active on the GGSN for pdn.aaaa.com from the 10.88.0.0 /24 network:

```
Router# show gprs gtp pdp-context all
TID                     MS_ADDR         Dynamic SGSN_addr       APN

6161616161610001        10.88.0.1       0       172.16.123.1    pdn.aaaa.com
6161616161610002        10.88.0.2       0       172.16.123.1    pdn.aaaa.com
6161616161610003        10.88.0.3       0       172.16.123.1    pdn.aaaa.com
6161616161610004        10.88.0.4       0       172.16.123.1    pdn.aaaa.com
6161616161610005        10.88.0.5       0       172.16.123.1    pdn.aaaa.com
```

Step 3 Use the **show ip route** command to verify that the corresponding aggregate route appears in the routing table.

Remember that a route only appears if the GGSN has received a PDP context from an MS on the network specified by the **gprs default aggregate** command (for any access point), or specified by the aggregate commands for a particular access point.

In our example, the following output for the **show ip route** command shows a single static route in the IP routing table for the GGSN, which routes the traffic for the 10.88.0.0/24 subnet through the virtual template (or Virtual-Access1) interface:

```
Router# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.80.0.0/16 is subnetted, 1 subnets
C       10.80.0.0 is directly connected, Loopback0
     10.113.0.0/16 is subnetted, 1 subnets
C       10.113.0.0 is directly connected, Virtual-Access1
     172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
C       172.16.43.192/28 is directly connected, FastEthernet0/0
S       172.16.43.0/24 is directly connected, FastEthernet0/0
S       172.16.43.35/32 is directly connected, Ethernet2/3
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
U       10.88.0.0/24 [1/0] via 0.0.0.0, Virtual-Access1
C       10.88.0.0/16 is directly connected, Loopback2
```

Step 4 Use the **show gprs access-point** command to verify whether route aggregation is configured on an APN, and if so, to display the aggregate ranges.

In the following output for the **show gprs access-point** command, an aggregate route has been configured for a particular access point, overriding the default aggregate routes configured globally:

```
Router#show gprs access-point 6
    apn_index 6          apn_name = gprs.corporate.com
    apn_mode:transparent
    apn_type:Real
    accounting:Disable
    wait_accounting:Disable
    input ACL:None, output ACL:None
    dynamic_address_pool:disable
```

```
apn_dhcp_server:0.0.0.0         backup:0.0.0.0
apn_dhcp_gateway_addr:0.0.0.0
apn_authentication_server_group:
apn_accounting_server_group:
apn_username:  apn_password:
subscribe_required:No
deactivate_pdp_context_on violation:No
network_activation_allowed:No
Block Foreign-MS Mode:Disable
VPN:Enable  (VRF Name :vpn1)
GPRS vaccess interface:Virtual-Access4
RADIUS attribute suppress MSISDN:Disabled
RADIUS attribute suppress IMSI:  Disabled
RADIUS attribute suppress SGSN Address:  Disabled
number of ip_address_allocated 0
idle timer:0
Security features
  Verify mobile source addr:     disable
  Verify mobile destination addr:disable

Total number of PDP in this APN :0

aggregate:
In APN:   55.0.0.0/8

In Global:40.40.0.0/16
             11.0.0.0/8
```

Configuration Examples

This section includes the following configuration examples for optimizing performance on the GGSN:

- CEF Switching Configuration Example, page 8-14
- Route Aggregation Configuration Example, page 8-16

CEF Switching Configuration Example

The following example enables CEF switching globally on the GGSN, which enables CEF for all interfaces on the GGSN. However, notice that CEF switching has been disabled at the tunnel0 interface. To support CEF switching, UDP checksum verification is disabled at the bottom of the configuration.

```
Current configuration : 4660 bytes
!
version 12.2
!
! Enable the router for GGSN services
!
service gprs ggsn
!
hostname myGGSN
!
ip dhcp pool 1
   network 10.60.0.0 255.255.255.240
   lease infinite
!
! Configure CEF switching globally
!
```

```
ip cef
!
interface Loopback0
 ip address 10.60.0.1 255.255.255.255
!
interface Tunnel0
 ip address 10.9.9.2 255.255.255.0
!
! Disable CEF switching on this
! interface only
!
 no ip route-cache cef
 tunnel source 10.25.25.1
 tunnel destination 10.25.25.2
!
interface Tunnel1
 ip address 10.11.11.2 255.255.255.0
 no ip mroute-cache
 tunnel source 10.26.26.1
 tunnel destination 10.26.26.2
!
interface FastEthernet0/0
 description Connect to Control subnet
 ip address 172.18.43.178 255.255.255.240
 no ip mroute-cache
 duplex full
!
interface FastEthernet1/0
 description - to PDN-2 over IPSec/GRE tunnel
 ip address 10.25.25.1 255.255.255.0
 no ip mroute-cache
 duplex full
!
interface Ethernet2/0
 description Connect to Server GW
 ip address 10.59.59.1 255.255.0.0
 duplex half
!
interface Ethernet2/1
 no ip address
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet2/3
 description - to CG - Ga
ip address 10.65.65.3 255.255.255.0
 duplex half
!
interface FastEthernet3/0
 description - to SGSN-1 Gn
 ip address 10.15.15.1 255.255.0.0
 no ip mroute-cache
 duplex full
!
interface FastEthernet5/0
 description - to PDN-3 over IPSec/GRE tunnel
 ip address 10.26.26.1 255.255.255.0
 no ip mroute-cache
 duplex full
!
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
!
ip default-gateway 172.18.43.177
ip classless
ip route 10.5.5.0 255.255.255.0 Tunnel0
ip route 10.30.30.0 255.255.255.0 10.25.25.2
ip route 192.168.0.0 255.255.0.0 172.18.43.177
ip route 172.0.0.0 255.0.0.0 172.18.43.177
ip route 172.18.43.35 255.255.255.255 10.59.59.3
ip route 192.168.220.1 255.255.255.255 FastEthernet3/0
no ip http server
!
! Configure the GGSN access point list
!
gprs access-point-list gprs
  access-point 1
   access-point-name gprs.cisco.com
   dhcp-server 10.60.0.1
   dhcp-gateway-address 10.60.0.1
   exit
  !
  access-point 2
   access-point-name hprs.cisco.com
   access-mode non-transparent
   ip-address-pool radius-client
   aaa-group authentication foo
   exit
   !
! GGSN global configuration parameters
!
gprs maximum-pdp-context-allowed 45000
gprs qos map canonical-qos
gprs default ip-address-pool dhcp-proxy-client
gprs default charging-gateway 10.65.65.2
!
! Disable verification of the UDP checksum
!
gprs gtp ip udp ignore checksum
!
radius-server host 172.18.43.7 auth-port 1645 acct-port 1646 non-standard
radius-server host 10.5.5.6 auth-port 1812 acct-port 1812 non-standard
radius-server host 172.18.61.17 auth-port 1645 acct-port 1645 non-standard key gociscogo
radius-server retransmit 1
radius-server timeout 1
radius-server key lab
call rsvp-sync
```

Route Aggregation Configuration Example

The following example shows a portion of a configuration that implements automatic route aggregation at an access point.

Notice that the **aggregate auto** command is configured at the access point where DHCP is being used. The **dhcp-gateway-address** command specifies the subnet addresses to be returned by the DHCP server. This address should match the IP address of a loopback interface on the GGSN. In addition, to accommodate route aggregation for another subnet 10.80.0.0, the **gprs default aggregate** global configuration command is used.

In this example, the GGSN aggregates routes for dynamically derived addresses for MSs through access point 8 based upon the address and mask returned by the DHCP server. For PDP context requests received for statically derived addresses on the 10.80.0.0 network, the GGSN also implements an aggregate route into its routing table, as configured by the **gprs default aggregate** command.

```
Current configuration :3815 bytes
!
version 12.2

...

service gprs ggsn

...
!
! Configures a loopback interface
! for the DHCP gateway address
!
interface Loopback0
ip address 10.80.0.1 255.255.255.255
!
gprs access-point-list gprs
 access-point 8
  access-point-name pdn.aaaa.com
!
! Enables DHCP services at the
! access point
!
  ip-address-pool dhcp-proxy-client
!
! Enables automatic route aggregation
!
  aggregate auto
!
! Configures an external DHCP server
! to support dynamic IP addressing for
! MSs through this access point
!
  dhcp-server 172.16.43.35
!
! Configures the subnet for which
! the DHCP server should return IP addresses
!
  dhcp-gateway-address 10.88.0.1
  exit
!
! Enables a single route to be established
! for PDP contexts with statically derived
! addresses on the 10.80.0.0 network
!
gprs default aggregate 10.80.0.0 255.255.255.0
```

Configuring QoS on the GGSN

This chapter describes how to configure Quality of Service (QoS) functions to differentiate traffic flow through the GGSN.

For a complete description of the GGSN commands in this chapter, refer to the *Cisco IOS Mobile Wireless 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

The Cisco IOS GGSN Release 4.0 software supports 2.5G GPRS QoS (as defined in GSM standards 02.60, 03.60, and 04.08) and 3G UMTS QoS (as defined in 3GPP X). Each GPRS/UMTS PDP context request contains either a GPRS QoS profile or UMTS QoS profile.

The implementation of QoS support in the GPRS/UMTS 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 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

GPRS QoS profiles is 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 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 Release 4.0 provides for the GPRS and UMTS QoS classes.

Configuring GPRS QoS on the GGSN

GGSN Release 3.0 and later supports 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

GGSN Release 1.2 and later supports 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, see 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 resource to be used for QoS processing. The GGSN allocates a portion of this total available resource for canonical QoS upon PDP context activation, based upon 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 resource 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)# gprs qos map canonical-qos`	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 Type of Service (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)# gprs canonical-qos map tos [premium tos-value [normal tos-value [best-effort tos-value]]]`	(Optional) Specifies a QoS mapping from the canonical QoS classes to an IP ToS precedence value, where *tos-value* 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 upon 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 10000 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)#` **`gprs canonical-qos gsn-resource-factor`** *`resource-factor`*	(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 resource to be reserved for best effort QoS classes on the GGSN 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)# gprs canonical-qos best-effort bandwidth-factor bandwidth-factor`	(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:

$$EB = Min[p, m + a (p - m)]$$

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)# gprs canonical-qos premium mean-throughput-deviation deviation-factor`	(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 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 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
!
. . .
!
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

GGSN Release 3.0 and later supports 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)# gprs qos map delay`	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 Type of Service (ToS) precedence. You can specify a mapping from the class1, class2, class3, 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 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)# gprs delay-qos map tos class1 tos-value [class2 tos-value [class3 tos-value [class-best-effort tos-value]]]`	(Optional) Specifies a QoS mapping from the delay QoS classes to an IP ToS precedence value, where *tos-value* 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 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 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
!
. . .
!
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

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)
Characteristics	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.
Example Applications	Voice over IP	Streaming audio and video	Web browsing	Downloading email
Diffserv Class / Map to DSCP	Expedited Forwarding Class	Assured Forwarding 2 Class	Assured Forwarding 3 Class	Best Effort

GGSN Release 4.0 and later supports end-to-end UMTS QoS by implementing it using 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, see the *Cisco IOS Quality of Service Solutions Configuration Guide*.

UMTS QoS Configuration Task List

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

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

- Enabling UMTS QoS Mapping on the GGSN (Required)
- Mapping UMTS QoS Traffic to a DiffServ PHB (Optional)
- Assigning DSCP to a DiffServ PHB Group (Optional)
- Configuring the DSCP in the Subscriber Datagram (Optional)
- Verifying UMTS QoS Configuration

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)# gprs qos map umts`	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, see 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
`Router(config)# gprs umts-qos map traffic-class traffic-class diffserv-phb-group`	Enables mapping of UMTS QoS traffic classes to a DiffServ PHB, where the UMTS traffic classes are: • signalling • conversational • streaming • interactive • background and the DiffServ PHB groups are: • signalling-class • ef-class • af1-class • af2-class • af3-class • af4-class • best-effort

Assigning a Differentiated Services Code Point

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
`Router(config)# gprs umts-qos map diffserv-phb diffserv-phb-group [dscp1] [dscp2] [dscp3]`	Assigns a DSCP to a DiffServ PHB group where the DiffServ PHB groups are: • signalling • ef-class • af1-class • af2-class • af3-class • af4-class • best-effort and the DSCPs are: • dscp1—Required for all classes. Specifies one of 64 DSCP values from 0 to 63. This DSCP value corresponds to drop precedence 1. • dscp2—(Optional for AF classes) Specifies one of 64 DSCP values from 0 to 63. This DSCP value corresponds to drop precedence 2. • dscp3—(Optional for AF classes) Specifies one of 64 DSCP values from 0 to 63. This DSCP value corresponds to drop precedence 3.

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
`Router(config)# gprs umts-qos dscp unmodified [up | down | all]`	Specifies that the subscriber datagram be forwarded through the GTP path without modifying its DSCP.

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

Verifying the UMTS QoS Configuration

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 Looback 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 maximum-pdp-context-allowed 200001
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
```

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 message. 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)# gprs qos default-response requested`	(Optional) Specifies that the GGSN sets its default QoS values in the response message exactly as requested in the create PDP context request message.

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.

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

This section provides a summary list of the **show** commands that you can use to monitor GPRS QoS or 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# **show gprs gtp pdp-context imsi** *hex-data*	Displays PDP contexts by International Mobile Subscriber Identity (IMSI).
Router# **show gprs gtp pdp-context qos-delay** {**class1** | **class2** | **class3** | **classbesteffort**}	Displays PDP contexts for a specified delay class type. Applies to GPRS QoS only.
Router# **show gprs gtp pdp-context qos-precedence** {**low** | **normal** | **high**}	Displays PDP contexts for a specified precedence type. Apply to GPRS QoS only.
Router# **show gprs gtp pdp-context tid** *hex-data*	Displays PDP contexts by tunnel ID.
Router# **show gprs gtp pdp-context umts-class** {**conversational** | **streaming** | **interactive** | **background**}	Displays PDP context by UMTS QoS traffic class. Apply to UMTS QoS only.
Router# **show gprs qos status**	Displays QoS statistics for the GGSN.
Router# **show gprs umts-qos map traffic-class**	Displays UMTS QoS mapping information.

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

To display GPRS QoS information for a particular PDP context, you can use the **show gprs gtp pdp-context** command using the **tid** or **imsi** keywords. 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 111111111111111
TID              MS Addr         Source  SGSN Addr       APN
111111111111111 10.0.0.1        Static  10.39.39.1      www.corporate.com

    current time: Nov 02 2001 15:36:42
    user_name (IMSI): 111111111111111    MS address: 10.2.0.1
    MS International PSTN/ISDN Number (MSISDN): 1111111111111
    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
    mnrgflag:          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 only apply 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 9-21
- Interpreting the Requested and Negotiated GPRS QoS, page 9-22
- Interpreting the Effective Bandwidth for a PDP Context, page 9-23 (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 to 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 to 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_neq 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 ÷ 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 20000 bps (a cumulative value, which corresponds to an effective bandwidth of 10000 bps for each PDP context in this example):

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: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 20000 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
1111111111111111 10.8.8.1        Static  10.39.39.1      gprt.cisco.com
2222222222222222 10.8.8.2        Static  10.39.39.1      gprt.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
3333333333333333 10.8.8.4        Static  10.39.39.1      gprt.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 GPRS QoS Information for a PDP Context, page 9-21
- Displaying GPRS QoS Status on the GGSN, page 9-23

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 using the **tid** or **imsi** keywords. 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 111111111111111
TID              MS Addr         Source  SGSN Addr       APN
111111111111111 10.0.0.1        Static  10.39.39.1      www.corporate.com

current time :Nov 12 2002 08:10:23
    user_name (IMSI):213000000000000    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_up:     0
    seq_tpdu_down:   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
    mnrgflag:         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 Example, page 9-27
- Delay QoS Configuration Example, page 9-29
- UMTS QoS Configuration Example, page 9-30

Canonical QoS Configuration Example

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 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 gprt.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 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
!
. . .
!
end
```

Delay QoS Configuration Example

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 gprt.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
!
. . .
!
end
```

UMTS QoS Configuration Example

The following example shows part of a sample GGSN configuration with the UMTS QoS method is 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 Looback 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 maximum-pdp-context-allowed 200001
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
```

Configuring Security on the GGSN

This chapter describes how to configure security features on the GGSN, including AAA, RADIUS, and IPSec.

The security configuration procedures and examples in this publication (aside from those related to GGSN-specific implementation) describe the basic commands that you can use to implement the security services.

For more detailed information about AAA, RADIUS, and IPSec security services in the Cisco IOS software, refer to the *Cisco IOS Security Configuration Guide* and *Cisco IOS Security Command Reference* publications.

For a complete description of the GGSN commands in this chapter, refer to the *Cisco IOS Mobile Wireless 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 Security Support on the GGSN

The GGSN supports many of the same levels of security that are available through the Cisco IOS software on the router, including the following types of security:

- Authentication, authorization, and accounting (AAA) network security services and server groups
- RADIUS security services
- IP Security Protocol (IPSec)

In addition, the GGSN software provides the ability to configure additional security features such as the following:

- Address verification
- Mobile-to-mobile traffic redirection
- IP access lists

AAA and RADIUS support provides the security services to authenticate and authorize access by mobile users to the GGSN and its APNs. IPSec support allows you to secure your data between the GGSN and its associated peers.

In some cases, such as with AAA and IPSec support, the GGSN works with the standard Cisco IOS software configuration without requiring configuration of any additional GGSN commands.

In the case of RADIUS server configuration, the GGSN requires that you enable AAA security and establish RADIUS server communication globally on the router. From there, you can configure RADIUS security for all GGSN access points, or on a per-access-point basis, using new GGSN configuration commands.

Note In addition to the AAA, RADIUS, and IPSec security services, the GGSN also supports IP access lists to further control access to APNs. The Cisco IOS GGSN software implements the new **ip-access-group** access-point configuration command to apply IP access list rules at an APN.

AAA Server Group Support

The Cisco Systems GGSN supports authentication and accounting at APNs using AAA server groups. By using AAA server groups, you gain the following benefits:

- You can selectively implement groups of servers for authentication and accounting at different APNs.
- You can configure different server groups for authentication services and accounting services in the same APN.
- You can control which RADIUS services you want to enable at a particular APN, such as AAA accounting.

For GTP-PPP termination and GTP-PPP regeneration on the GGSN, transparent access mode is used to allow PPP to perform the appropriate AAA functions; however, you can still configure AAA server groups to specify the corresponding server groups for AAA support.

The GGSN supports the implementation of AAA server groups at both the global and access-point configuration levels. You can minimize your configuration by specifying the configuration that you want to support across most APNs, at the global configuration level. Then, at the access-point configuration level, you can selectively modify the services and server groups that you want to support at a particular APN. Therefore, you can override the AAA server global configuration at the APN configuration level.

To configure a default AAA server group to be used for all APNs on the GGSN, use the **gprs default aaa-group** global configuration command. To specify a different AAA server group to be used at a particular APN for authentication or accounting, use the **aaa-group** access-point configuration command.

If authentication is enabled on the APN, then the GGSN first looks for an authentication server group at the APN. If an authentication server group is not found at the APN, then the GGSN looks for a globally configured, GPRS/UMTS default authentication server group.

If accounting is enabled on the APN, then the GGSN looks for an accounting server group at the APN or globally in the following order:

- First, at the APN for an accounting server group—configured in the **aaa-group accounting** command.
- Second, for a global GPRS/UMTS default accounting server group—configured in the **gprs default aaa-group accounting** command.
- Third, at the APN for an authentication server group—configured in the **aaa-group authentication** command.
- Last, for a global GPRS/UMTS default authentication server group—configured in the **gprs default aaa-group authentication** command.

To complete the configuration, you also must specify the following configuration elements on the GGSN:

- Configure the RADIUS servers using the **radius-server host** command.
- Define a server group with the IP addresses of the AAA servers in that group using the **aaa group server** global configuration command.
- Enable the type of AAA services (accounting and authentication) to be supported on the APN.
  - The GGSN enables accounting by default for non-transparent APNs.

    You can disable accounting services at the APN using the **aaa-accounting disable** command.
  - You can enable authentication at the APN level by configuring the **access-mode non-transparent** command. When you enable authentication, the GGSN automatically enables accounting on the APN. There is not a global configuration command to enable or disable authentication.
- Configure AAA accounting and authentication using the **aaa accounting** and **aaa authentication** global configuration commands.

Note For more information about AAA and RADIUS global configuration commands, see the *Cisco IOS Security Command Reference.*

Configuring AAA Security Globally

Authentication, authorization, and accounting (AAA) network security services provide the primary framework through which you set up access control on your GGSN. This section provides information about the basic commands used to implement AAA security on a Cisco Systems' router.

To enable AAA and configure authentication and authorization, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA globally.
Step 2	`Router(config)# aaa authentication ppp {default | list-name} method1 [method2...]`	Creates a local authentication method list, with the following options: • **default**—Specifies that the authentication methods that follow this argument are the default list of authentication methods when a user logs in to the router. • *method*—Specifies a valid AAA authentication method for PPP. For example, **group RADIUS** enables global RADIUS authentication.
Step 3	`Router(config)# aaa authorization {auth-proxy | network | exec | commands level | reverse-access} {default | list-name} [method1 [method2...]]`	Creates an authorization method list for a particular authorization type and enables authorization.
Step 4	`Router(config)# aaa accounting {system default [vrf vrf-name] | network {default | none | start-stop | stop-only | wait-start} group group-name`	Enables AAA accounting of requested services for billing or security purposes when you use RADIUS.

For more information about configuring AAA, refer to the *Cisco IOS Security Configuration Guide* and *Cisco IOS Security Command Reference* publications.

Configuring RADIUS Server Communication Globally

This section describes how to configure a global RADIUS server host that the GGSN can use to authenticate and authorize users. You can configure additional RADIUS server communication at the GPRS global configuration level.

To globally configure RADIUS server communication on the router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# radius-server host {hostname | ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]`	Specifies the IP address or host name of the remote RADIUS server host. The following options are available: • **auth-port**—Specifies the UDP destination port for authentication requests. • **acct-port**—Specifies the UDP destination port for accounting requests. • **timeout**—Specifies the time interval (in the range1 to 1000 seconds) that the router waits for the RADIUS server to reply before retransmitting. This setting overrides the global value of the **radius-server timeout** command. If no timeout value is specified, the global value is used. • **retransmit**—Specifies the number of times (in the range 1 to 100) a RADIUS request is resent to a server, if that server is not responding or is responding slowly. This setting overrides the global value of the **radius-server retransmit** command. • **key**—Specifies the authentication and encryption key used between the router and the RADIUS daemon running on this RADIUS server. This setting overrides the global value of the **radius-server key** command.
Step 2	`Router(config)# radius-server key string`	Specifies the shared secret text string used between the router and the vendor-proprietary RADIUS server. The router and the RADIUS server use this text string to encrypt passwords and exchange responses.

For more information about configuring RADIUS security, refer to the *Cisco IOS Security Configuration Guide* and *Cisco IOS Security Command Reference* publications.

For an example, see the "RADIUS Server Global Configuration Example" section on page 10-28.

Note Although you can configure the **radius-server host** command multiple times, the Cisco IOS software supports only one RADIUS server at the same IP address.

Configuring RADIUS Server Communication at the GPRS/UMTS Configuration Level

To complete the security configuration for the GGSN, you must configure non-transparent access for each access point. When you configure security at the GPRS global configuration level, you also can configure RADIUS server communication for all access points or for a specific access point.

Configuring RADIUS at the GPRS global configuration level includes the following tasks:

- Configuring Non-Transparent Access Mode, page 10-6 (Required)
- Specifying a AAA Server Group for All Access Points, page 10-7 (Optional)
- Specifying a AAA Server Group for a Particular Access Point, page 10-8 (Optional)
- Configuring AAA Accounting Services at an Access Point, page 10-8 (Optional)

Configuring Non-Transparent Access Mode

To support RADIUS authentication on the GGSN, you must configure the GGSN access points for non-transparent access. You must configure non-transparent access for every access point at which you want to support RADIUS services. There is not a way to globally specify the access mode.

Note For GTP-PPP termination and GTP-PPP regeneration on the GGSN, transparent access mode is used to allow PPP to perform the appropriate AAA functions; however, you can still configure AAA server groups to specify the corresponding server groups for AAA support.

To configure non-transparent access for a GGSN access point, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gprs access-point-list list-name`	Specifies the access-point list name and enters access-point list configuration mode.
Step 2	`Router(config-ap-list)# access-point access-point-index`	Specifies the number associated with an existing access point definition (or creates a new access point), and enters access point configuration mode.
Step 3	`Router(config-access-point)# access-mode non-transparent`	Specifies that the GGSN requests user authentication at the access point to a PDN.

For more information about configuring GGSN access points, see the “Configuring Access Points on the GGSN” section on page 6-6.

Specifying a AAA Server Group for All Access Points

After you have configured RADIUS server communication at the global level, you can configure a default AAA server group to be used by all GGSN access points.

To specify a default AAA server group for all GGSN access points, use the following command in global configuration mode:

Command	Purpose
`Router(config)# gprs default aaa-group {authentication | accounting} server-group`	Specifies a default AAA server group and assigns the type of AAA services to be supported by the server group for all access points on the GGSN, where: • **authentication**—Assigns the selected server group for authentication services on all APNs. • **accounting**—Assigns the selected server group for accounting services on all APNs. • *server-group*—Specifies the name of a AAA server group to be used for AAA services on all APNs. **Note** The name of the AAA server group that you specify must correspond to a server group that you configure using the **aaa group server** command.

Specifying a AAA Server Group for a Particular Access Point

To override the default AAA server group configured for all access points, you can specify a different AAA server group for a particular access point. Or, if you choose not to configure a default AAA server group, you can specify a AAA server group at each access point.

To specify a AAA server group for a particular access point, use the following command in access-point configuration mode:

Command	Purpose
`Router(config-access-point)# aaa-group {authentication | accounting} server-group`	Specifies a default AAA server group and assigns the type of AAA services to be supported by the server group for a particular access point on the GGSN, where: • **authentication**—Assigns the selected server group for authentication services on the APN. • **accounting**—Assigns the selected server group for accounting services on the APN. • *server-group*—Specifies the name of a AAA server group to be used for AAA services on the APN. **Note** The name of the AAA server group that you specify must correspond to a server group that you configure using the **aaa group server** command.

Configuring AAA Accounting Services at an Access Point

The Cisco Systems GGSN has different defaults for enabling and disabling accounting services for transparent and non-transparent access points:

- If you configure an APN for non-transparent access using the **access-mode** command, the GGSN automatically enables accounting with authentication at the APN.
- If you configure an APN for transparent access, which is the default access mode, the GGSN automatically disables accounting at the APN.

Therefore, if you have configured a transparent access APN and you want to provide accounting at that APN, you need to configure the **aaa-accounting enable** command at the APN.

However, for accounting to occur, you also must complete the configuration by specifying the following other configuration elements on the GGSN:

- Enable AAA services using the **aaa new-model** global configuration command.
- Define a server group with the IP addresses of the RADIUS servers in that group using the **aaa group server** global configuration command.

- Configure the following AAA services:
  - AAA authentication using the **aaa authentication** global configuration command
  - AAA authorization using the **aaa authorization** global configuration command
  - AAA accounting using the **aaa accounting** global configuration command
- Assign the type of services that the AAA server group should provide. If you only want the server group to support accounting services, then you need to configure the server for accounting only. You can assign the AAA services to the AAA server groups either at the GPRS global configuration level using the **gprs default aaa-group** command, or at the APN using the **aaa-group** command.
- Configure the RADIUS servers using the **radius-server host** command.

Note For more information about AAA and RADIUS global configuration commands, see the *Cisco IOS Security Command Reference.*

To selectively disable accounting at specific APNs where you do not want that service, use the **aaa-accounting disable** access-point configuration command.

There is not a **no** form of this command.

You can verify whether AAA accounting is enabled or disabled at an APN using the **show gprs access-point** command.

To enable or disable accounting at an access point, use the following command in access-point configuration mode:

Command	Purpose
`Router(config-access-point)# aaa-accounting {enable | disable}`	Enables or disables accounting for a particular access point on the GGSN.

Configuring Additional RADIUS Security Services

This section describes how to configure RADIUS security services that the GGSN can use to authenticate and authorize users.

This section includes the following tasks:

Configuring the MSISDN IE for RADIUS Requests

To specify that the first byte of the Mobile Station International PSTN/ISDN (MSISDN) information element is included in a RADIUS request, use the following command beginning in global configuration mode:

Command	Purpose
`Router(config)# gprs radius msisdn first-byte`	Specifies that the first byte of the MSISDN IE is included in a RADIUS request.

Configuring the Vendor-Specific Attribute for RADIUS Requests

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information to the RADIUS server by using the vendor-specific attribute (attribute 26). Vendor-specific attributes (VSAs) makes a larger set of information available for communication by allowing vendors to support their own extended attributes not suitable for general use.

Table 10-1 lists and describes the Third Generation Partnership Project (3GPP) VSA sub-attributes that the GGSN can send in authentication and accounting requests to a RADIUS server.

Table 10-1 3GPP VSA Sub-Attributes

Number	Vendor-Proprietary Attribute	Description
1	3GPP-IMSI	International Mobile Subscriber Identity (IMSI) number for a user. This sub-attribute can be suppressed using the **radius attribute suppress imsi** command.
2	3GPP-Charging-Id	Charging ID for this PDP context.
3	3GPP-PDP-Type	Type of PDP context (for example, IP or PPP).
4	3GPP-CG-Address	IP address of the current active charging gateway. If there is no current active charging gateway, GGSN sends 0.0.0.0.
5	3GPP-GPRS-QoS-Profile	QoS negotiated values. This sub-attribute can be suppressed using the **radius attribute suppress qos** command.
6	3GPP-SGSN-Address	IP address of the SGSN that is used by the GTP control plane for handling control messages. This address might be used to identify the PLMN to which the user is attached. This sub-attribute can be suppressed using the **radius attribute suppress sgsn-address** command.
7	3GPP-GGSN-Address	IP address of the GGSN that is used by the GTP control plane for the context establishment. This address is the same as the GGSN IP address used in G-CDRs.

Table 10-1 3GPP VSA Sub-Attributes

Number	Vendor-Proprietary Attribute	Description
8	3GPP-IMSI-MCC-MNC	MCC and MNC extracted from the user's IMSI number (the first 5 or 6 digits depending on the IMSI). This sub-attribute requires that the MCC and MNC values that the GGSN uses be configured using the **gprs mcc mnc** global configuration command.
9	3GPP-GGSN-MCC-MNC	MCC and MNC of the network to which the GGSN belongs. This sub-attribute requires that the MCC and MNC values that the GGSN uses be configured using the **gprs mcc mnc** global configuration command.
12	3GPP-Selection-Mode	Selection mode for this PDP context received in the Create PDP Context request.

To configure the GGSN to send and recognize VSAs as defined by RADIUS attribute 26, use the following command in global configuration mode:

Command	Purpose
`Router(config)#radius-server vsa send [accounting | authentication]`	(Optional) Enables the GGSN to send and recognized VSAs as defined by RADIUS IETF attribute 26.

For more information on configuring the use of vendor-specific attributes, refer to the *Cisco IOS Security Configuration Guide* and *Cisco IOS Security Command Reference* publications.

Suppressing Attributes for RADIUS Authentication

You can configure the GGSN to suppress certain attributes in its authentication and accounting request to a RADIUS server. The following sections describe the attributes you can suppress and how to do so.

The following topics are included in this section:

Suppressing the MSISDN Number for RADIUS Authentication

Certain countries have privacy laws which prohibit service providers from identifying the MSISDN number of mobile stations in authentication requests. Use the **msisdn suppression** command to specify a value that the GGSN sends in place of the MSISDN number in its authentication requests to a RADIUS server. If no value is configured, then no number is sent to the RADIUS server.

To use the **msisdn suppression** command, you must configure a RADIUS server either globally or at the access point and specify non-transparent access mode.

To specify that the GGSN overrides or suppresses the MSISDN number in its RADIUS authentication, use the following command beginning in access-point configuration mode:

Command	Purpose
`Router(config-access-point)# msisdn suppression [value]`	(Optional) Specifies that the GGSN overrides the MSISDN number with a pre-configured value in its authentication requests to a RADIUS server.

Suppressing the 3GPP-IMSI VSA Sub-Attribute for RADIUS Authentication

To specify that the GGSN suppress the Third Generation Partnership Project (3GPP) vendor-specific attribute (VSA) 3GPP-International Mobile Subscriber Identity (3GPP-IMSI) number in its authentication and accounting requests to a RADIUS server, use the **radius attribute suppress imsi** access point configuration command.

To specify that the GGSN suppress the 3GPP VSA 3GPP-IMSI number in its authentication and accounting requests to a RADIUS server, use the following command beginning in access-point configuration mode:

Command	Purpose
`Router(config-access-point)# radius attribute suppress imsi`	(Optional) Specifies that the GGSN suppresses the 3GPP-IMSI number in its authentication and accounting requests to a RADIUS server.

Suppressing the 3GPP-GPRS-QoS Profile VSA Sub-Attribute for RADIUS Authentication

To specify that the GGSN suppress the 3GPP-GPRS-Qos Profile in its authentication and accounting requests to a RADIUS server, use the **radius attribute suppress qos** access point configuration command.

To specify that the GGSN suppress the 3GPP-GPRS-Qos Profile in its authentication and accounting requests to a RADIUS server, use the following command beginning in access-point configuration mode:

Command	Purpose
`Router(config-access-point)# radius attribute suppress qos`	(Optional) Specifies that the GGSN suppresses the 3GPP-GPRS-Qos Profile in its authentication and accounting requests to a RADIUS server.

Suppressing the 3GPP-GPRS-SGSN-Address VSA Sub-Attribute for RADIUS Authentication

To specify that the GGSN suppress the 3GPP-GPRS-SGSN-Address in its authentication and accounting requests to a RADIUS server, use the **radius attribute suppress sgsn-address** access point configuration command.

To specify that the GGSN suppress the 3GPP-GPRS-SGSN-Address in its authentication and accounting requests to a RADIUS server, use the following command beginning in access-point configuration mode:

Command	Purpose
`Router(config-access-point)# radius attribute suppress sgsn-address`	(Optional) Specifies that the GGSN suppresses the 3GPP-GPRS-SGSN-Address in its authentication and accounting requests to a RADIUS server.

Obtaining DNS and NetBIOS Address Information from a RADIUS Server

To obtain DNS address and NetBIOS address information from a RADIUS server, issue the following command in global configuration mode:

Command	Purpose
`Router(config)#radius-server vsa send [accounting | authentication]`	(Optional) Enables the GGSN to send and recognized VSAs as defined by RADIUS IETF attribute 26.

Note For the DNS and NetBIOS address information to be sent to an MS, the dynamic address allocation method using an IP address pool supplied by a RADIUS server must be configured for the access point using the **ip-address-pool radius-client** command. For more information about configuring an access point, see the "Configuring Access Points on the GGSN" section on page 6-6.

Configuring the GGSN to Wait for a RADIUS Response

Use the **gtp response-message wait-accounting** command to configure the GGSN to wait for a RADIUS accounting response from the RADIUS accounting server, before sending a create PDP context response to the SGSN.

If the GGSN does not receive a response from the RADIUS accounting server when you have configured the **gtp response-message wait-accounting** command, then the GGSN rejects the PDP context request.

The GGSN supports configuration of RADIUS response message waiting at both the global and access-point configuration levels. You can minimize your configuration by specifying the configuration that you want to support across most APNs, at the global configuration level. Then, at the access-point configuration level, you can selectively modify the behavior that you want to support at a particular APN. Therefore, at the APN configuration level, you can override the global configuration of RADIUS response message waiting.

To configure the GGSN to wait for a RADIUS accounting response as the default behavior for all APNs, use the **gprs gtp response-message wait-accounting** global configuration command. To disable this behavior for a particular APN, use the **no gtp response-message wait-accounting** access-point configuration command.

To verify whether RADIUS response message waiting is enabled or disabled at an APN, you can use the **show gprs access-point** command and observe the value reported in the wait_accounting output field.

To configure the GGSN to wait for a RADIUS accounting response globally, use the following command in global configuration mode:

Command	Purpose
`Router(config)# gprs gtp response-message wait-accounting`	Configures the GGSN to wait for a RADIUS accounting response before sending a create PDP context response to the SGSN, for create PDP context requests received across all access points.

To configure the GGSN to wait for a RADIUS accounting response for a particular access point, use the following command in access-point configuration mode:

Command	Purpose
`Router(config-access-point)# gtp response-message wait-accounting`	Configures the GGSN to wait for a RADIUS accounting response before sending a create PDP context response to the SGSN, for create PDP context requests received at a particular access point.

Configuring Access to a RADIUS Server Using VRF

The Cisco IOS GGSN software supports access to a RADIUS server using VRF. This Cisco IOS software feature is called *Per VRF AAA* and using this feature, ISPs can partition AAA services base on VRF. This permits the GGSN to communicate directly with the customer RADIUS server associated with the customer VPN without having to go through a RADIUS proxy. Thus, ISPs can scale their VPN offerings more efficiently because they no longer need to proxy AAA to provide their customers the flexibility demanded.

To support this configuration, AAA must be VRF aware. ISPs must define multiple instances of the same operational parameters—such as AAA server groups, method lists, system accounting, and protocol-specific parameters—and secure the parameters to the VRF partitions.

If an AAA configuration, such as a method list, is uniquely defined many times, the specification of an AAA server that is based on IP addresses and port numbers might create an overlapping of private addresses between VRFs. Securing AAA method lists to a VRF can be accomplished from one or more of the following sources:

- Virtual Template—Used as a generic interface configuration.
- Service Provider AAA server—Used to associate a remote user with a specific VPN based on the domain name or Dialed Number Identification Service (DNIS). The server then provides the VPN-specific configuration for the virtual access interface, which includes the IP address and port number of the customer AAA server.
- Customer VPN AAA server—Used to authenticate the remote user and to provide user-specific configurations for the virtual access interface.

Note Global AAA accounting configurations and some AAA protocol-specific parameters cannot be logically grouped under the Virtual Template configuration.

When configuring the Per VRF feature, keep in mind the following:

- To prevent possible overlapping of private addresses between VRFs, AAA servers must be defined in a single global pool that is to be used in the server groups.
- Servers can no longer be uniquely identified by IP addresses and port numbers.
- "Private" servers (servers with private addresses within the default server group that contains all the servers) can be defined within the server group and remain hidden from other groups. The list of servers in server groups includes references to the hosts in the global configuration as well as the definitions of private servers.

Note If private server parameters are not specified, global configurations are used. If global configurations are not specified, default values are used.

- All server operational parameters can be configured per host, per server group, or globally. Per-host configurations have precedence over per-server group configurations. Per-server group configurations have precedence over global configurations.

Note For complete information on configuring access to a RADIUS server using VRF, refer to the *Per VRF AAA* feature module.

This section describes configuring and establishing access to a private RADIUS server using VRF. For global RADIUS services, ensure that you have configured a globally located server.

To configure access to a RADIUS server using VRF, complete the following tasks:

- Enabling AAA Globally, page 10-16 (Required)
- Configuring a VRF-Aware Private RADIUS Server Group, page 10-16 (Required)
- Configuring Accounting, Authentication, and Authorization Using Named Method Lists, page 10-17 (Required)
- Configuring a VRF Routing Table, page 10-17 (Required)
- Configuring VRF on an Interface, page 10-18 (Required)
- Configuring VRF under an Access Point for Access to the Private RADIUS Server, page 10-18 (Required)
- Configuring a Route to the RADIUS Server Using VRF, page 10-19 (Optional)

Enabling AAA Globally

If AAA has not been enabled globally on the GGSN, you will need to do so before configuring access to a private RADIUS server via VRF.

To enable AAA globally, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA globally.

Configuring a VRF-Aware Private RADIUS Server Group

To configure private server operational parameters, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa group server radius group-name`	Groups different RADIUS server hosts into distinct lists and distinct methods. • *group-name*—Character string used to name the group of servers.
Step 2	`Router(config-sg-radius)# server-private ip-address auth-port port_num acct-port port_num key string`	Configures the IP address of the private RADIUS server for the group server. • *ip-address*—Specifies the IP address of the private RADIUS server host. • **auth-port** *port_num*—Specifies a port solely for authentication. • **acct-port** *port_num*—Specifies a port solely for accounting. • *string*—(Optional) Specifies the authentication and encryption key for all RADIUS communications between the router and the RADIUS server. **Note** If private server parameters are not specified, global configurations are used. If global configurations are not specified, default values are used.
Step 3	`Router(config-sg-radius)# ip vrf forwarding vrf-name`	Configures the VRF reference of the AAA RADIUS server group. • *vrf-name*—Name assigned to a VRF.

Configuring Accounting, Authentication, and Authorization Using Named Method Lists

To configure AAA authorization using named method lists, perform the following tasks beginning in global configuration mode:

Step 4	`Router(config)# aaa authentication ppp {default | list-name} method1 [method2...]`	Creates a local authentication method list, with the following options: • **default**—Specifies that the authentication methods that follow this argument are the default list of authentication methods when a user logs in to the router. • *method*—Specifies a valid AAA authentication method for PPP. For example, **group RADIUS** enables global RADIUS authentication.
Step 5	`Router(config)# aaa authorization {auth-proxy | network | exec | commands level | reverse-access} {default | list-name} [method1 [method2...]]`	Creates an authorization method list for a particular authorization type and enables authorization.
Step 6	`Router(config)# aaa accounting {system default [vrf vrf-name] | network {default | none | start-stop | stop-only | wait-start} group group-name`	Enables AAA accounting of requested services for billing or security purposes when you use RADIUS.

Configuring a VRF Routing Table

To configure a VRF routing table on the GGSN for access to the Private RADIUS server, use the following command beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip vrf vrf-name`	Configures a VRF routing table and enters VRF configuration mode.
Step 2	`Router(config-vrf)# rd route-distinguisher`	Creates routing and forwarding tables for a VRF and specifies the default route distinguisher for a VPN.

Configuring VRF on an Interface

To configure VRF on a physical interface to the PDN using Fast Ethernet over the Gi interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type slot/port`	Defines a physical interface on the GGSN, where *type* is **fastethernet**, and *slot/port* is the hardware slot and port on the interface.
Step 2	`Router(config-if)# ip vrf forwarding vrf-name`	Associates a VRF with an interface or subinterface. **Note** The *vrf-name* argument should match the name of the VRF that you configured using the **ip vrf** command in the "Configuring Accounting, Authentication, and Authorization Using Named Method Lists" section on page 10-17. **Note** The IP address defined on the interface will get removed when you associate a VRF with the interface. Therefore, you will need to reconfigure the IP address for the interface.
Step 3	`Router(config-if)# ip address ip-address mask`	Specifies an IP address for the interface, where: • *ip-address*—Specifies the IP address of the interface in dotted decimal format. • *mask*—Specifies a subnet mask in dotted decimal format.

Configuring VRF under an Access Point for Access to the Private RADIUS Server

To configure VRF under an access point to use to access a RADIUS server, use the following commands beginning in access-point list configuration mode:

	Command	Purpose
Step 1	`Router(config-ap-list)# access-point access-point-index`	Specifies an index number for a new access point definition, or references an existing access point definition, and enters access point configuration mode.
Step 2	`Router(config-access-point)# access-point-name apn-name`	Specifies the network (or domain) name for a PDN that users can access from the GGSN at a defined access point. **Note** The *apn-name* must match the APN that has been provisioned at the MS, HLR, and DNS server.

	Command	Purpose
Step 3	`Router(config-access-point)# aaa-group authentication server-group`	Specifies a default AAA server group and assigns the type of AAA services to be supported by the server group for a particular access point on the GGSN, where: • **authentication**—Assigns the selected server group for authentication services on the APN. • *server-group*—Specifies the name of a AAA server group to be used for AAA services on the APN. **Note** The name of the AAA server group that you specify must correspond to a server group that you configure using the **aaa group server** command.
Step 4	`Router(config-access-point)# access-mode non-transparent`	Specifies for the GGSN to act as a proxy for authentication.
Step 5	`Router(config-access-point)# ip-address-pool radius-client`	Specifies for the RADIUS server to provide the IP address pool for the current access point. **Note** If you are using a dynamic address allocation method, then you must configure this command according to the appropriate IP address pool source.
Step 6	`Router(config-access-point)# vrf vrf-name`	Configures VPN routing and forwarding at a GGSN access point and associates the access point with a particular VRF instance. **Note** The *vrf-name* argument should match the name of the VRF that you configured using the **ip vrf** command in the "Configuring Accounting, Authentication, and Authorization Using Named Method Lists" section on page 10-17.

Configuring a Route to the RADIUS Server Using VRF

Be sure a route exists between the VRF instance and the RADIUS server. You can verify connectivity by using the **ping** command from the VRF to the RADIUS server. To configure a route, you can use a static route or a routing protocol.

Configuring a Static Route Using VRF

To configure a static route using, use the following command beginning in global configuration mode:

Command	Purpose
`Router(config)# ip route vrf vrf-name prefix mask [next-hop-address] [interface {interface-number}] [global] [distance] [permanent] [tag tag]`	Configures a static IP route, where: • *vrf-name*—Specifies the name of the VPN routing/forwarding instance (VRF) for the static route. • *prefix*—Specifies the IP route prefix for the destination. • *mask*—Specifies the prefix mask for the destination. • *next-hop-address*—Specifies the IP address of the next hop that can be used to reach the destination network. • *interface interface-number*—Specifies the network interface type and interface number that can be used to reach the destination network. • **global**—Specifies that the given next hop address is in the non-VRF routing table. • *distance*—Specifies an administrative distance for the route. • **tag** *tag*—Specifies a tag value that can be used as a "match" value for controlling redistribution via route maps. • **permanent**—Specifies that the route will not be removed, even if the interface shuts down.

Verifying a Static Route Using VRF

To verify that the static VRF route that you configured, use the **show ip route vrf** privileged EXEC command as shown in the following example:

```
Router# show ip route vrf vpn1 static

      172.16.0.0/16 is subnetted, 1 subnets
C       172.16.0.1 is directly connected, Ethernet5/1
C       10.100.0.3/8 is directly connected, Virtual-Access5
```

Configuring an OSPF Route Using VRF

To configure an OSPF route using VRF, use the following command beginning in global configuration mode:

Command	Purpose
`Router(config)# router ospf process-id [vrf vrf-name]`	Enables OSPF routing, and enters router configuration mode, where, • *process-id*—Specifies an internally used identification parameter for an OSPF routing process. The *process-id* is locally assigned and can be any positive integer. A unique value is assigned for each OSPF routing process. • **vrf** *vrf-name*—Specifies the name of the VPN routing/forwarding instance.

Configuring IPSec Network Security

In Cisco IOS Release 12.1(5)T and later, the GGSN software supports the IP security protocol for data authentication, confidentiality, encryption and integrity. IPSec data security can be implemented between the GGSN and another router on the PDN.

Note To support IPSec on the GGSN, you must install an ISA card on your router.

Configuring IPSec network security includes the following tasks:

- Configuring an IKE Policy, page 10-21 (Required)
- Configuring Pre-Shared Keys, page 10-23 (Required, when pre-shared authentication is configured)
- Configuring Transform Sets, page 10-24 (Optional)
- Configuring Crypto Map Entries that Use IKE to Establish Security Associations, page 10-24 (Optional)

For more information about configuring IPSec, refer to the *Cisco IOS Security Configuration Guide* and *Cisco IOS Security Command Reference* publications.

For an example, see the "IPSec Configuration Example" section on page 10-31.

Configuring an IKE Policy

You can create multiple Internet Key Exchange (IKE) policies, each with a different combination of parameter values. For each policy that you create, you assign a unique priority (1 through 10,000, with 1 being the highest priority).

You can configure multiple policies on each peer—but at least one of these policies must contain exactly the same encryption, hash, authentication, and Diffie-Hellman parameter values as one of the policies on the remote peer. For example, you can configure multiple policies on the GGSN to correlate with the policies for different PDNs.

To configure an IKE policy on the GGSN and corresponding PDN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto isakmp policy priority`	Identifies the IKE policy, where *priority* is an integer (from 1 to 10,000) that uniquely identifies the policy. This command enters you into ISAKMP policy configuration mode.
Step 2	`Router(config-isakmp)# encryption des`	Specifies the encryption algorithm, where: • **des**—Specifies 56-bit Data Encryption Standard (DES)-Cipher Block Chaining (CBC). This is the default value. **Note** Triple DES, or 168-bit DES encryption is supported in the Cisco IOS software. It can be configured by using this command and specifying the **3des** optional keyword. GGSN Release 1.4 in Cisco IOS Release 12.2 does not support the **3des** optional keyword.
Step 3	`Router(config-isakmp)# hash {sha | md5}`	Specifies the hash algorithm, where: • **sha**—Specifies the Secure Hash Algorithm (SHA)-1. This is the default value. • **md5**—Specifies the Message Digest 5 hash algorithm.
Step 4	`Router(config-isakmp)# authentication {rsa-sig | rsa-encr | pre-share}`	Specifies the authentication method, where: • **rsa-sig**—Specifies the public key encryption system developed by Ron Rivest, Adi Shamir, and Leonard Adleman, which provides non-repudiation. This is the default value. • **rsa-encr**—Specifies RSA encrypted nonces, which provide repudiation. • **pre-share**—Specifies a pre-shared key that does not require use of a certification authority. Pre-shared keys might be easier to configure in a small network with less than 10 nodes. RSA signatures can be considered more secure than pre-shared keys. If you configure **pre-share** authentication, then you must configure the pre-shared keys on both the local and remote peer (GGSN and PDN).

	Command	Purpose
Step 5	`Router(config-isakmp)# group {1 | 2}`	Specifies the Diffie-Hellman group identifier, where: • **1**—Specifies 768-bit Diffie-Hellman. This is the default value. • **2**—Specifies 1024-bit Diffie-Hellman. **Note** The 1024-bit Diffie-Hellman option is harder to crack, but requires more CPU time to execute.
Step 6	`Router(config-isakmp)# lifetime seconds`	Specifies the security association's lifetime (in seconds). The default value is 86,400 seconds (1 day).

For more information about the meaning of the IKE policy parameters, refer to the *Cisco IOS Security Configuration Guide* and *Cisco IOS Security Command Reference* publications.

Configuring Pre-Shared Keys

When you configure the **pre-share** authentication method for your IKE policy, you also must configure the pre-shared keys on the GGSN and remote peer, or PDN.

To configure pre-shared keys on the GGSN and corresponding PDN, use one of the following commands beginning in global configuration mode:

Command	Purpose

`Router(config)# crypto isakmp key keystring address peer-address`
or
`Router(config)# crypto isakmp key keystring hostname peer-hostname`
	Specifies the shared key to be used between a local peer (GGSN) and particular remote peer (PDN).
	If the remote peer, or PDN, specifies the ISAKMP identity with an address, use the **address** keyword; otherwise use the **hostname** keyword.
	When configuring the pre-shared keys on the GGSN, use the address or hostname of the PDN. When configuring the pre-shared keys on the PDN, use the address or hostname of the GGSN.

Configuring Transform Sets

A transform set represents a certain combination of security protocols and algorithms. During the IPSec security association negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

To configure a transform set on the GGSN and corresponding PDN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto ipsec transform-set transform-set-name transform1 [transform2 [transform3]]`	Defines a transform set and enters crypto transform configuration mode. There are complex rules defining which entries you can use for the transform arguments. For more information, refer to the *Cisco IOS Security Configuration Guide* and *Cisco IOS Security Command Reference* publications.
Step 2	`Router(config-crypto-transform)# mode [tunnel | transport]`	(Optional) Changes the mode associated with the transform set. The following options are available: • **tunnel**—Protects (encrypts, authenticates) and encapsulates the entire original IP packet • **transport**—Protects (encrypts, authenticates) and encapsulates the payload or data portion of the IP packet. **Note** The mode setting is applicable only to traffic whose source and destination addresses are the IPSec peer addresses; it is ignored for all other traffic.

Configuring Crypto Map Entries that Use IKE to Establish Security Associations

When you use IKE to establish security associations, you can specify a list of acceptable settings to be used during IPSec peer negotiation using a crypto map entry.

To configure crypto map entries on the GGSN and corresponding PDN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name seq-num ipsec-isakmp`	Creates or modifies a crypto map entry and enters crypto map configuration mode.
Step 2	`Router(config-crypto-map)# match address access-list-id`	Names an extended access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec in the context of the current crypto map entry.
Step 3	`Router(config-crypto-map)# set peer {hostname | ip-address}`	Specifies a remote IPSec peer. This is the peer to which IPSec-protected traffic can be forwarded.

	Command	Purpose
Step 4	`Router(config-crypto-map)# set transform-set transform-set-name1 [transform-set-name2...transform-set-name6]`	Specifies which transform sets are allowed for this crypto map entry. List multiple transform sets in order of priority (highest priority first).

Step 5	`Router(config-crypto-map)# set security-association lifetime seconds seconds`
	and/or
	`set security-association lifetime kilobytes kilobytes`
		(Optional) Specifies a security association lifetime for the crypto map entry, if you want the security associations for the current crypto map entry to be negotiated using different IPSec security association lifetimes than the global lifetimes.

Step 6	`Router(config-crypto-map)# set security-association level per-host`	(Optional) Specifies that separate security associations should be established for each source/destination pair. **Note** Use this command with care, as multiple streams between given subnets can rapidly consume resources.
Step 7	`Router(config-crypto-map)# set pfs [group1 | group2]`	(Optional) Specifies that IPSec should ask for perfect forward secrecy (PFS) when requesting new security associations for the current crypto map entry, or should demand PFS in requests received from the IPSec peer.
Step 8	`Router(config-crypto-map)# exit`	Exits crypto map configuration mode.
Step 9	`Router(config)# interface fastethernet slot/port`	Accesses the Gi interface to which you want to apply the cryto map.
Step 10	`Router(config-if)# crypto map map-name`	Applies the crypto map set to the interface.

Securing the GGSN Mobile (Gn) Interface

With GGSN 3.0 on Cisco IOS Release 12.2(8) B or later, features have been added to provide additional security for the GGSN mobile interface against attacks that can lead to illegal access to a network or even network downtime.

Configuring Address Verification

Use the **security verify source** access point configuration command to configure the GGSN to verify the source IP address of an upstream TPDU against the address previously assigned to an MS.

When the **security verify source** command is configured on an APN, the GGSN verifies the source address of a TPDU before GTP will accept and forward it. If the GGSN determines that the address differs from that previously assigned to the MS, it drops the TPDU and accounts it as an illegal packet in its PDP context and APN. Configuring the **security verify source** access point configuration command protects the GGSN from faked user identities.

Use the **security verify destination** access point configuration command to have the GGSN verify the destination addresses of upstream TPDUs against global lists of PLMN addresses specified using the **gprs plmn ip address** command. If the GGSN determines that a destination address of a TPDU is within the range of a list of addresses, it drops the TPDU. If it determines that the TPDU contains a destination address that does not fall within the range of a list, it forwards the TPDU to its final destination.

Note The **security verify destination** command is not applied to APNs using VRF. In addition, the verification of destination addresses does not apply to GTP-PPP regeneration or GTP-PPP with L2TP.

To configure address verification for a GGSN access point, use the following commands beginning in access-point configuration mode:

Command	Purpose
`Router(config-access-point)# security verify {source | destination}`	(Optional) Specifies that the GGSN verify the source or destination address in TPDUs received from a Gn interface.

Note Both the verification of destination addresses and source addresses can be configured on an APN.

Configuring Mobile-to-Mobile Traffic Redirection

Mobile-to-mobile traffic enters and exits through a Gn interface. Therefore, it is switched by the GGSN without ever going through a Gi interface on the network side. Because of this, firewalls deployed on the network side of a GGSN do not have an opportunity to verify this level of traffic.

Use the **redirect intermobile ip** access-point command to redirect mobile-to-mobile traffic to an external device (such as an external firewall) for verification.

Command	Purpose
`Router(config-access-point)# redirect intermobile ip ip address`	(Optional) Specifies that mobile-to-mobile traffic be redirected to an external device.

Note Redirection of intermobile traffic does not occur on an ingress APN unless the TPDUs are exiting the same APN. In addition, redirection of TPDUs tunneled by L2TP from the ingress APN to the LNS of the PDN does not occur.

Configuration Examples

This section includes the following configuration examples for security on the GGSN:

AAA Security Configuration Example

The following example shows how to enable AAA security globally on the router, and specify global RADIUS authentication and authorization:

```
! Enables AAA globally
aaa new-model
!
! Creates a local authentication list for use on
! serial interfaces running PPP using RADIUS
!
aaa authentication ppp foo group foo
!
! Enables authorization and creates an authorization
! method list for all network-related service requests
! and enables authorization using a RADIUS server
!
aaa authorization network network foo group foo
```

For more information about configuring AAA, refer to the *Cisco IOS Security Configuration Guide* and *Cisco IOS Security Command Reference* publications.

RADIUS Server Global Configuration Example

The following example shows how to globally configure RADIUS server communication on the router:

```
! Specifies a global RADIUS server host at IP address 10.100.0.2
! Port 1645 is destination port for authentication requests
! Port 1646 is the destination port for accounting requests
! Specifies the key "foo" for this radius host only
!
radius-server host 10.100.0.2 auth-port 1645 acct-port 1646 key foo
!
! Sets the authentication and encryption key to mykey for all
! RADIUS communications between the router and the RADIUS daemon
!
radius-server key mykey
```

Note Although you can configure the **radius-server host** command multiple times, the Cisco IOS software supports only one RADIUS server at the same IP address.

For more information about configuring RADIUS security, refer to the *Cisco IOS Security Configuration Guide* and *Cisco IOS Security Command Reference* publications.

RADIUS Server Group Configuration Example

The following configuration example defines four AAA server groups on the GGSN: foo, foo1, foo2, and foo3, shown by the **aaa group server** commands.

Using the **gprs default aaa-group** command, two of these server groups are globally defined as default server groups: foo2 for authentication, and foo3 for accounting.

At access-point 1, which is enabled for authentication, the default global authentication server group of foo2 is overridden and the server group named foo is designated to provide authentication services on the APN. Notice that accounting services are not explicitly configured at that access point, but are automatically enabled because authentication is enabled. Because there is a globally defined accounting server-group defined, the server named foo3 will be used for accounting services.

At access-point 4, which is enabled for accounting using the **aaa-accounting enable** command, the default accounting server group of foo3 is overridden and the server group named foo1 is designated to provide accounting services on the APN.

Access-point 5 does not support any AAA services because it is configured for transparent access mode.

```
! Enables AAA globally
!
aaa new-model
!
! Defines AAA server groups
!
aaa group server radius foo
 server 10.2.3.4 auth-port 1645 acct-port 1646
 server 10.6.7.8 auth-port 1645 acct-port 1646
aaa group server radius foo1
 server 10.10.0.1 auth-port 1645 acct-port 1646
aaa group server radius foo2
 server 10.2.3.4 auth-port 1645 acct-port 1646
 server 10.10.0.1 auth-port 1645 acct-port 1646
aaa group server foo3
 server 10.6.7.8 auth-port 1645 acct-port 1646
 server 10.10.0.1 auth-port 1645 acct-port 1646
```

```
!
! Configures AAA authentication
! and authorization
!
aaa authentication ppp foo group foo
aaa authentication ppp foo2 group foo2
aaa authorization network foo group foo
aaa accounting network foo start-stop group foo
aaa accounting network foo1 start-stop group foo1
aaa accounting network foo2 start-stop group foo2
aaa accounting network foo3 start-stop group foo3
!
gprs access-point-list gprs
 access-point 1
  access-mode non-transparent
  access-point-name www.pdn1.com
!
! Specifies a RADIUS server group
! for use by the GGSN to authenticate
! mobile users at this access point
!
  aaa-group authentication foo
!
 access-point 4
  access-point-name www.pdn2.com
!
! Enables AAA accounting services
!
  aaa-accounting enable
!
! Specifies a RADIUS server group
! for use by the GGSN for accounting
! services at this access point

  aaa-group accounting foo1
!
 access-point 5
  access-point-name www.pdn3.com
!
! Configures default AAA server
! groups for the GGSN for authentication
! and accounting services
!
gprs default aaa-group authentication foo2
gprs default aaa-group accounting foo3
!
! Configures global RADIUS server hosts
! and specifies destination ports for
! authentication and accounting requests
!
radius-server host 10.2.3.4 auth-port 1645 acct-port 1646 non-standard
radius-server host 10.6.7.8 auth-port 1645 acct-port 1646 non-standard
radius-server host 10.10.0.1 auth-port 1645 acct-port 1646 non-standard
radius-server key ggsntel
```

Note Although you can configure the **radius-server host** command multiple times, the Cisco IOS software supports only one RADIUS server at the same IP address.

RADIUS Response Message Configuration Example

The following example globally configures the GGSN to wait for a RADIUS accounting response from the RADIUS server before sending a create PDP context response to the SGSN. The GGSN waits for a response for PDP context requests received across all access points, except access-point 1. RADIUS response message waiting has been overridden at access-point 1 using the **no gtp response-message wait-accounting** command:

```
! Enables AAA globally
!
aaa new-model
!
! Defines AAA server group
!
aaa group server radius foo
 server 10.2.3.4 auth-port 1645 acct-port 1646
 server 10.6.7.8 auth-port 1645 acct-port 1646
!
! Configures AAA authentication
! and authorization
!
aaa authentication ppp foo group foo
aaa authorization network foo group foo
aaa accounting network foo start-stop group foo
!
gprs access-point-list gprs
 access-point 1
  access-mode non-transparent
  access-point-name www.pdn1.com
  aaa-group authentication foo
!
! Disables waiting for RADIUS response
! message at APN 1
!
  no gtp response-message wait-accounting
  exit
access-point 2
 access-mode non-transparent
 access-point-name www.pdn2.com
 aaa-group authentication foo
!
! Enables waiting for RADIUS response
! messages across all APNs (except APN 1)
!
gprs gtp response-message wait-accounting
!
! Configures global RADIUS server hosts
! and specifies destination ports for
! authentication and accounting requests
!
radius-server host 10.2.3.4 auth-port 1645 acct-port 1646 non-standard
radius-server host 10.6.7.8 auth-port 1645 acct-port 1646 non-standard
radius-server key ggsntel
```

IPSec Configuration Example

IP Security Protocol is configured between two peers to establish data security services. For GPRS/UMTS, IPSec configuration is applicable between the GGSN and a router on a PDN. The following example shows configuration of IPSec on the GGSN and an associated PDN, including the complete global and GGSN configuration commands:

GGSN configuration

```
!
hostname ggsn1
!
enable password ggsn1password
!
ip host pdn1a 10.58.0.8
!
ip dhcp-server 10.40.0.3
ip dhcp-server 10.100.0.3
ip address-pool dhcp-proxy-client
!
! IPSec configuration for GGSN
crypto isakmp policy 1
  authentication pre-share
  group 2
!
! 10.58.0.8 is address of peer, or PDN
!
crypto isakmp key sharedkey address 10.58.0.8
crypto ipsec transform-set auth2 esp-des esp-sha-hmac
crypto map test 10 ipsec-isakmp
  set peer 10.58.0.8
  set transform-set auth2
  match address 133
!
! ISA card is required for IPSec support
!
controller ISA 1/1
!
interface loopback 1
  ip address 10.7.7.7 255.255.255.0
!
interface Ethernet5/0
  description TFTP D0OWNLOAD
  ip address 10.103.0.7 255.255.0.0
!
interface FastEthernet0/0
  description CONNECT TO sgsn-a
  ip address 10.56.0.7 255.255.0.0
!
interface FastEthernet4/0
  description CONNECT TO Gi
  ip address 10.58.0.7 255.255.0.0
  crypto map test
!
interface Virtual-Template1
  ip unnumber loopback 1
  encapsulation gtp
  ip mroute-cache
  gprs access-point-list abc
!
```

```
router eigrp 10
  network 10.2.0.0
  network 10.56.0.0
  network 10.58.0.0
!
!
ip route 10.5.5.5 255.255.255.255 FastEthernet0/0
!
access-list 133 permit ip 10.56.0.0 0.0.255.255 10.59.0.0 0.0.255.255
access-list 133 permit ip 10.2.0.0 0.0.255.255 10.59.0.0 0.0.255.255
!
!
gprs access-point-list abc
  access-point 1
    access-point-name apn1.cisco.com
    dhcp-server 10.100.0.3
  exit
!
access-point 2
  access-point-name apn2.cisco.com
  dhcp-server 10.100.0.3
  exit
!
access-point 3
  access-point-name www.apn3.com
  dhcp-server 10.100.0.3
  exit
!
!
gprs default charging-gateway 10.58.0.4 10.58.0.2
gprs charging server-switch-timer 0
```

PDN configuration

```
hostname pdn1a
!
enable password pdn1apassword
!
ip host ggsn1 10.58.0.7
!
!
! IPSec configuration on the PDN
!
crypto isakmp policy 1
  authentication pre-share
  group 2
!
! 10.58.0.7 is address of peer, or GGSN
!
crypto isakmp key sharedkey address 10.58.0.7
crypto ipsec transform-set auth2 ah-sha-hmac esp-des esp-sha-hmac
crypto map test 10 ipsec-isakmp
  set peer 10.58.0.7
  set transform-set auth2
  match address 144
!
!
controller ISA 1/1
!
interface Ethernet5/0
  description TFTP DOWNLOAD
  ip address 10.103.0.8 255.255.0.0
  ip helper-address 10.100.0.3
```

```
!
interface FastEthernet0/0
  description CONNECT TO Intranet
  ip address 10.59.0.8 255.255.0.0
!
interface FastEthernet4/0
  description CONNECT TO Gi
  ip address 10.58.0.8 255.255.0.0
  crypto map test

!
!
! ISA card is required for IPSec support
router eigrp 10
  network 10.2.0.0
  network 10.58.0.0
  network 10.59.0.0
!
!
access-list 144 permit ip 10.59.0.0 0.0.255.255 10.56.0.0 0.0.255.255
access-list 144 permit ip 10.59.0.0 0.0.255.255 10.2.0.0 0.0.255.255
!
!
```

Address Verification and Mobile-to-Mobile Traffic Redirection Example

The following example shows how to enable address verification on the router and specify that mobile-to-mobile traffic be redirected to an external device:

```
! Defines PLMN address ranges
gprs plmn ip address 1.1.1.1 1.1.1.99
gprs plmn ip address 1.1.2.1 1.1.2.49
!
! Enters access-point configuration mode
! and turns on source and destination address
! verification and mobile-to-mobile traffic redirection
!
gprs access-point-list gprs
     access-point 1
         access-point-name www.abc.com
         security verify source
         security verify destination
         redirection intermobile ip 10.1.1.1
!
```

Access to a Private RADIUS Server Using VRF Configuration Example

The following examples shows an example of configuring access to a private RADIUS server using VRF:

```
! Enables AAA globally
aaa new-model
!
! Configures a VRF-Aware Private RADIUS Server Group named vrf_aware_radius
!
aaa group server radius vrf_aware_radius
 server-private 99.100.0.2 auth-port 1645 acct-port 1646 key cisco
 ip vrf forwarding vpn4
!
! Configures Authentication, Authorization, and Accounting using named method lists
```

```
!
aaa authentication ppp vrf_aware_radius group vrf_aware_radius
aaa authorization network default local group radius
aaa authorization network vrf_aware_radius group vrf_aware_radius
aaa accounting network vrf_aware_radius start-stop group vrf_aware_radius
aaa session-id common
!
! Configures a VRF routing table
!
ip vrf vpn4
 rd 104:1
!
! Configures VRF on an interface
!
interface FastEthernet0/0
 ip vrf forwarding vpn4   <=== new
 ip address 99.108.0.4 255.255.255.0
!
! Configures VRF on an access point for access to the server
!
access-point 17
   access-point-name radius_vrf
   access-mode non-transparent
   aaa-group authentication vrf_aware_radius
   aaa-group accounting vrf_aware_radius
   ip-address-pool radius-client
   vrf vpn4
   exit
```

Configuring DHCP on the GGSN

This chapter describes how to configure the Dynamic Host Configuration Protocol (DHCP) on the GGSN. The GGSN uses DHCP to assign IP addresses to mobile station users who need to access the PDN.

For a complete description of the GGSN commands in this chapter, refer to the *Cisco IOS Mobile Wireless 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 Configuring DHCP on the GGSN

You can use local DHCP services within the Cisco IOS software, or you can configure the GGSN to use an external DHCP server such as the Cisco Network Registrar (CNR). For information about configuring internal DHCP services in the Cisco IOS software, see the *Cisco IOS Configuration Fundamentals Configuration Guide*.

The DHCP server can be specified in two ways:

- At the global configuration level, using the **gprs default dhcp-server** command.
- At the access-point configuration level, using the **dhcp-server** command.

To configure DHCP support on the GGSN, you must configure either the **gprs default ip-address-pool** global configuration command or the **ip-address-pool** access-point configuration command with the **dhcp-proxy-client** keyword option.

After you configure the access point for DHCP proxy client services, use the **dhcp-server** access-point configuration command to specify a DHCP server.

Use the *ip-address* argument to specify the IP address of the DHCP server. The second, optional *ip-address* argument can be used to specify the IP address of a backup DHCP server to be used in the event that the primary DHCP server is unavailable. If you do not specify a backup DHCP server, then no backup DHCP server is available.

If you specify a DHCP server at the access-point level using the **dhcp-server** command, then the server address specified at the access point overrides the address specified at the global level. If you do not specify a DHCP server address at the access-point level, then the address specified at the global level is used.

Therefore, you can have a global address setting and also one or more local access-point level settings if you need to use different DHCP servers for different access points.

Use the **vrf** keyword when the DHCP server itself is located within the address space of a VRF interface on the GGSN. If the DHCP server is located within the VRF address space, then the corresponding loopback interface for the **dhcp-gateway-address** must also be configured within the VRF address space.

Configuring DHCP Server Communication Globally

This section describes how to configure a global DHCP server host that the GGSN can use to assign IP addresses to mobile users. You can configure additional DHCP server communication at the GGSN global configuration level.

To globally configure DHCP server communication on the router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip address-pool {dhcp-proxy-client | local}`	Specifies an IP address pool mechanism, where: • **dhcp-proxy-client**—Specifies the router as the proxy-client between a third-party DHCP server and peers connecting to the router. • **local**—Specifies the local address pool named "default". **Note** There is no default option for the **ip address-pool** command. If you configure a local address pool using the **local** keyword, you can also configure the optional commands in Step 4 and Step 5.
Step 2	`Router(config)# ip dhcp-server {ip-address | name}`	Specifies the IP address or name of a DHCP server.
Step 3	`Router(config)# ip dhcp excluded address low-address [high-address]`	(Optional) Specifies IP addresses that a DHCP server should not assign to DHCP clients, where: • *low-address*—Specifies the first IP address in an excluded address range. This address is typically the address of the DHCP server itself. • *high-address*—(Optional) Specifies the last IP address in the excluded address range.

	Command	Purpose
Step 4	`Router(config)# ip dhcp pool name`	(Optional—Supports **ip address-pool local** command only.) Configures a DHCP address pool and enters DHCP pool configuration mode, where *name* can be either a symbolic string (such as "engineering") or an integer (such as 0).
Step 5	`Router(config-dhcp)# network network-number [mask | /prefix-length]`	(Optional—Supports **ip address-pool local** command only.) Specifies the subnet network number and mask of the DHCP address pool. The prefix length specifies the number of bits that comprise the address prefix. The prefix is an alternative way of specifying the network mask of the client. The prefix length must be preceded by a forward slash (/).

For more information about configuring global DHCP services, refer to the *Cisco IOS IP Configuration Guide*, *Cisco IOS IP Command References*, and the *Cisco IOS Dial Technologies Command Reference* publications.

Configuring DHCP at the GGSN Global Configuration Level

To complete the DHCP configuration for the GGSN, you can configure DHCP at the GGSN global configuration level. When you configure DHCP at the GGSN configuration level, you can configure DHCP server communication for all access points, or for a specific access point.

Configuring DHCP at the GGSN configuration level includes the following tasks:

- Configuring a Loopback Interface, page 11-3 (Required)
- Specifying a DHCP Server for All Access Points, page 11-4 (Optional)
- Specifying a DHCP Server for a Particular Access Point, page 11-6 (Optional)

Configuring a Loopback Interface

When you configure a DHCP gateway address for DHCP services at an access point, and when you are supporting unique supernets across all access points on the GGSN for DHCP, then you must configure a loopback interface for each unique network.

A loopback interface is a software-only interface that emulates an interface that is always up. It is a virtual interface supported on all platforms. The interface-number is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces you can create.

To configure a loopback interface on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface loopback interface-number`	Defines a loopback interface on the GGSN, where *interface-number* identifies the loopback interface.
Step 2	`Router(config-if)# ip address ip-address mask [secondary]`	Specifies an IP address for the interface, where: • *ip-address*—Specifies the IP address of the interface in dotted decimal format. • *mask*—Specifies a subnet mask in dotted decimal format. • **secondary**—Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address. **Note** The *ip-address* corresponds to the IP address of the DHCP gateway address at the access point. The mask should be 255.255.255.255 to match the **dhcp-gateway-address** value exactly.

Specifying a DHCP Server for All Access Points

When processing DHCP address allocation, the GGSN software first checks to see whether a DHCP server has been specified at the access-point configuration level. If so, it uses the DHCP server specified at the access point. If no DHCP server is specified at the access-point configuration level, then the GGSN uses the default GGSN DHCP server.

To specify a DHCP server for all GGSN access points, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gprs default ip-address-pool {dhcp-proxy-client | radius-client | disable}`	Specifies a dynamic address allocation method using IP address pools for the GGSN, where: • **dhcp-proxy-client**—Specifies that the GGSN dynamically acquires IP addresses for an MS from a DHCP server. Use this keyword to enable DHCP services. • **radius-client**—Specifies that the GGSN dynamically acquires IP addresses for an MS from a RADIUS server. • **disable**—Disables dynamic address allocation by the GGSN. There is no default option for this command.
Step 2	`Router(config)# gprs default dhcp-server {ip-address | name} [{ip-address | name}]`	Specifies a primary (and backup) DHCP server from which the GGSN obtains IP address leases for mobile users, where: • *ip-address*—Specifies the IP address of a DHCP server. The second (optional) *ip-address* argument specifies the IP address of a backup DHCP server. • *name*—Specifies the host name of a DHCP server. The second (optional) *name* argument specifies the host name of a backup DHCP server.

Specifying a DHCP Server for a Particular Access Point

To override the default DHCP server configured for all access points, you can specify a different DHCP server for a particular access point. Or, if you choose not to configure a default GGSN DHCP server, you can specify a DHCP server at each access point.

To specify a DHCP server for a particular access point, use the following commands beginning in access-point configuration mode:

	Command	Purpose
Step 1	`Router(config-access-point)# ip-address-pool {dhcp-proxy-client | radius-client | disable}`	Specifies a dynamic address allocation method using IP address pools for the current access point, where: • **dhcp-proxy-client**—Specifies that the access point IP address pool is maintained on a DHCP server. Use this keyword to enable DHCP services. • **radius-client**—Specifies that the access point IP address pool is allocated through a RADIUS server. • **disable**—Disables dynamic address allocation for the current access point. There is no default option for this command.
Step 2	`Router(config-access-point)# dhcp-server {ip-address} [ip-address] [vrf]`	Specifies a primary (and backup) DHCP server that the GGSN uses at a particular access point to obtain IP address leases for mobile users for access to a PDN, where: • *ip-address*—Specifies the IP address of a DHCP server. The second (optional) *ip-address* argument specifies the IP address of a backup DHCP server. • **vrf**—DHCP server uses the VPN routing and forwarding (VRF) table that is associated with the APN.
Step 3	`Router(config-access-point)# dhcp-gateway-address ip-address`	Specifies the subnet in which the DHCP server should return addresses for DHCP requests for MS users entering a particular PDN access point. **Note** You must configure a corresponding loopback interface with the same IP address as the DHCP gateway address.

Configuring a Local DHCP Server

Although most networks use external DHCP servers, such as that available through the Cisco Network Registrar (CNR), you can also configure internal DHCP services on the GGSN. If you use local DHCP services on the GGSN, then there are a couple of commands that you should configure to improve the internal DHCP response times.

To optimize local DHCP services on the GGSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip dhcp ping packets 0`	Specifies that the Cisco IOS DHCP Server sends 0 packets to a pool address as part of a ping operation.
Step 2	`Router(config)# ip dhcp ping timeout 100`	Specifies that the Cisco IOS DHCP Server waits for a ping reply from an address pool for 100 milliseconds.

Configuration Example

The following example shows a VRF configuration for vpn3 (without tunneling) using the **ip vrf** global configuration command. Because the **ip vrf** command establishes both VRF and CEF routing tables, notice that **ip cef** also is configured at the global configuration level to enable CEF switching at all of the interfaces.

The following other configuration elements must also associate the same VRF named vpn3:

- FastEthernet0/0 is configured as the Gi interface using the **ip vrf forwarding** interface configuration command.
- Access-point 2 implements VRF using the **vrf** command access-point configuration command.

The DHCP server at access-point 2 also is configured to support VRF. Notice that access-point 1 uses the same DHCP server, but is not supporting the VRF address space. The IP addresses for access-point 1 will apply to the global routing table:

```
aaa new-model
!
aaa group server radius foo
 server 10.2.3.4
 server 10.6.7.8
!
aaa authentication ppp foo group foo
aaa authorization network foo group foo
aaa accounting network foo start-stop group foo
!
ip cef
!
ip vrf vpn3
 rd 300:3
!
interface Loopback1
 ip address 10.30.30.30 255.255.255.255
!
interface Loopback2
 ip vrf forwarding vpn3
 ip address 10.27.27.27 255.255.255.255
!
```

```
interface FastEthernet0/0
 ip vrf forwarding vpn3
 ip address 10.50.0.1 255.255.0.0
 duplex half
!
interface FastEthernet1/0
 ip address 10.70.0.1 255.255.0.0
 duplex half
!
interface loopback 1
 ip address 10.8.0.1 255.255.255.0
!
interface Virtual-Template1
 ip unnumber loopback 1
 encapsulation gtp
 gprs access-point-list gprs
!
ip route 10.10.0.1 255.255.255.255 Virtual-Template1
ip route vrf vpn3 10.100.0.5 255.255.255.0 fa0/0 10.50.0.2
ip route 10.200.0.5 255.255.255.0 fa1/0 10.70.0.2
!
no ip http server
!
gprs access-point-list gprs
 access-point 1
  access-point-name gprs.pdn.com
  ip-address-pool dhcp-proxy-client
  dhcp-server 10.200.0.5
  dhcp-gateway-address 10.30.30.30
  network-request-activation
  exit
  !
 access-point 2
  access-point-name gprs.pdn2.com
  access-mode non-transparent
  ip-address-pool dhcp-proxy-client
  dhcp-server 10.100.0.5 10.100.0.6 vrf
  dhcp-gateway-address 10.27.27.27
  aaa-group authentication foo
  vrf vpn3
  exit
!
gprs default ip-address-pool dhcp-proxy-client
gprs gtp ip udp ignore checksum
!
radius-server host 10.2.3.4 auth-port 1645 acct-port 1646 non-standard
radius-server host 10.6.7.8 auth-port 1645 acct-port 1646 non-standard
radius-server key ggsntel
```

Configuring Load Balancing on the GGSN

This chapter describes how to configure a GGSN to support load balancing functions using the Cisco IOS software Server Load Balancing (SLB) feature. GGSN SLB provides increased reliability and availability when you are using multiple Cisco Systems GGSNs or non-Cisco GGSNs in your GPRS/UMTS network.

For a complete description of the GGSN commands in this chapter, refer to the *Cisco IOS Mobile Wireless Command Reference*. For a complete description of the other Cisco IOS SLB commands in this chapter, refer to the *IOS Server Load Balancing*, 12.1(13)E3 documentation located at Cisco.com at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121limit/121e/121e13/index.htm

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:

Note You can also use the GTP Director Module (GDM) to obtain simple, round-robin load balancing in environments where non-Cisco GGSNs are found. For more information about GDM and load balancing, see the *GTP Director Module* section of this book.

Overview of Load Balancing on the GGSN

This section provides an overview of the Cisco IOS SLB feature and load balancing support on the GGSN. It includes the following sections:

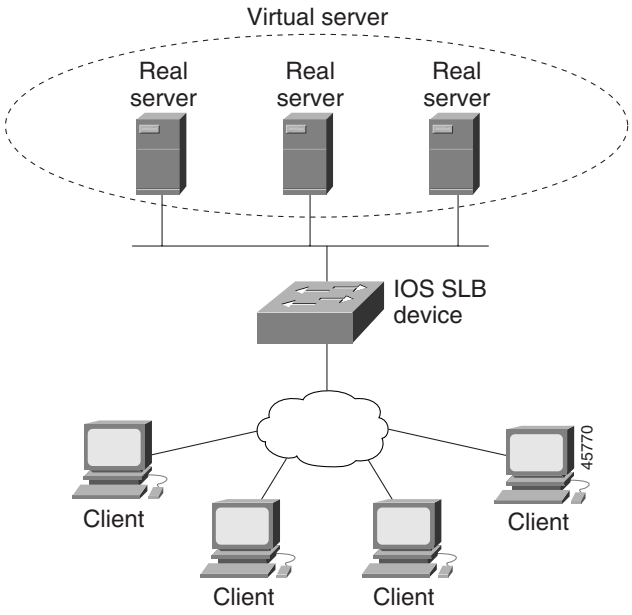
Overview of Cisco IOS SLB

The Cisco SLB feature is an IOS-based solution that provides IP server load balancing. Using the Cisco IOS SLB feature, you can define a *virtual server* that represents a group of *real servers* in a cluster of network servers known as a *server farm*. In this environment, the clients connect to the IP address of the virtual server. When a client initiates a connection to the virtual server, the Cisco IOS SLB feature chooses a real server for the connection based on a configured *load-balancing algorithm*.

The Cisco IOS SLB feature also provides firewall load balancing, which balances flows across a group of *firewalls* called a *firewall farm*.

Figure 12-1 illustrates a logical view of a simple Cisco IOS SLB network.

Figure 12-1 Logical View of IOS SLB



GGSN GTP Load Balancing Support

IOS SLB provides GGSN GTP load balancing and increased reliability and availability for the GGSN. GGSN GTP load balancing supports a subset of the overall server load balancing functions that are available in the Cisco IOS SLB feature. Therefore, the full scope of Cisco IOS SLB functions are not applicable to the GPRS/UMTS environment. For more information about unsupported functions, see the "Restrictions" section on page 12-6.

When configuring GTP load balancing, a pool of GGSNs is configured as a server farm in IOS SLB. These are the GGSNs across which you want to load balance GTP sessions. A virtual server instance is configured in IOS SLB to load balance GTP sessions across the GGSN farm. This virtual server is associated with the server farm that you configured in IOS SLB.

IOS SLB supports two types of GTP load balancing:

- GTP Load Balancing without GTP Cause Code Inspection, page 12-3
- GTP Load Balancing with GTP Cause Code Inspection, page 12-3

GTP Load Balancing without GTP Cause Code Inspection

GTP load balancing *without* GTP cause code inspection enabled is recommended for Cisco GGSNs. It has the following characteristics:

- Can operate in dispatched mode or in directed server NAT mode, but not in directed client NAT mode. In dispatched mode, the GGSNs must be Layer 2-adjacent to the IOS SLB device.
- Does not support stateful backup.
- Delivers tunnel creation messages destined to the virtual GGSN IP address to one of the real GGSNs, using the weighted round robin load-balancing algorithm. See the "Weighted Round Robin" section on page 12-4 for more information about this algorithm.
- Requires DFP to account for secondary PDP contexts in GTP v1.

GTP Load Balancing with GTP Cause Code Inspection

GTP load balancing *with* GTP cause code inspection enabled allows IOS SLB to monitor all PDP context signaling flows to and from GGSN server farms. This enables IOS SLB to monitor GTP failure cause codes, detecting system-level problems in both Cisco and non-Cisco GGSNs.

Table 1 lists the PDP create response cause codes and the corresponding actions taken by IOS SLB:

Table 1 PDP Create Response Cause Codes and Corresponding IOS SLB Actions

Cause Code	IOS SLB Action
Request Accepted	Establish session
No Resource Available	Fail current real, reassign session, drop the response
All dynamic addresses are occupied	Fail current real, reassign session, drop the response
No memory is available	Fail current real, reassign session, drop the response
System Failure	Fail current real, reassign session, drop the response
Missing or Unknown APN	Forward the response
Unknown PDP Address or PDP type	Forward the response
User Authentication Failed	Forward the response
Semantic error in TFT operation	Forward the response
Syntactic error in TFT operation	Forward the response
Semantic error in packet filter	Forward the response
Syntactic error in packet filter	Forward the response
Mandatory IE incorrect	Forward the response
Mandatory IE missing	Forward the response
Optional IE incorrect	Forward the response
Invalid message format	Forward the response
Version not supported	Forward the response

GTP load balancing *with* GTP cause code inspection enabled has the following characteristics:

- Must operate in directed server NAT mode.
- Assigns PDP context creates from a specific IMSI to the same GGSN.

- Supports stateful backup.
- Tracks the number of open PDP contexts for each GGSN, which enables GGSN server farms to use the weighted least connections (**leastconns**) algorithm for GTP load balancing. See the "Weighted Least Connections" section on page 12-4 for more information about this algorithm.
- Enables IOS SLB to deny access to a virtual GGSN if the carrier code of the requesting International Mobile Subscriber ID (IMSI) does not match a specified value.
- Enables IOS SLB to support secondary PDP contexts, even without DFP.

Weighted Round Robin

The weighted round robin algorithm specifies that the real server used for a new connection to the virtual server is chosen from the server farm in a circular fashion. Each real server is assigned a weight, *n*, that represents its capacity to handle connections, as compared to the other real servers associated with the virtual server. That is, new connections are assigned to a given real server *n* times before the next real server in the server farm is chosen.

For example, assume a server farm comprised of real server ServerA with $n = 3$, ServerB with $n = 1$, and ServerC with $n = 2$. The first three connections to the virtual server are assigned to ServerA, the fourth connection to ServerB, and the fifth and sixth connections to ServerC.

Note Assigning a weight of n=1 to all of the servers in the server farm configures the IOS SLB device to use a simple round robin algorithm.

GTP load balancing *without* GTP cause code inspection enabled requires the weighted round robin algorithm. A server farm that uses weighted least connections can be bound to a virtual server providing GTP load balancing without GTP cause code inspection enabled, but you cannot place the virtual server **INSERVICE**. If you try to do so, IOS SLB issues an error message.

Weighted Least Connections

When GTP cause code inspection is enabled, GTP load balancing supports the Cisco IOS SLB weighted least connections algorithm.

The weighted least connections algorithm specifies that the next real server chosen from a server farm for a new connection to the virtual server is the server with the fewest active connections. Each real server is assigned a weight for this algorithm, also. When weights are assigned, the server with the fewest connections is based on the number of active connections on each server, and on the relative capacity of each server. The capacity of a given real server is calculated as the assigned weight of that server divided by the sum of the assigned weights of all of the real servers associated with that virtual server, or $n_1/(n_1+n_2+n_3...)$.

For example, assume a server farm comprised of real server ServerA with $n = 3$, ServerB with $n = 1$, and ServerC with $n = 2$. ServerA would have a calculated capacity of 3/(3+1+2), or half of all active connections on the virtual server, ServerB one-sixth of all active connections, and ServerC one-third of all active connections. At any point in time, the next connection to the virtual server would be assigned to the real server whose number of active connections is farthest below its calculated capacity.

Note Assigning a weight of n=1 to all of the servers in the server farm configures the IOS SLB device to use a simple least-connection algorithm.

GTP load balancing *without* GTP cause code inspection enabled *does not* support the weighted least connections algorithm.

GTP load balancing *with* GTP cause code inspection *does* support the weighted least connections algorithm.

Dynamic Feedback Protocol for IOS SLB

In GTP load balancing, IOS SLB knows when a PDP context is established, but it does not know when PDP contexts are cleared, and therefore it cannot know the number of open PDP contexts for each GGSN. Use the IOS SLB Dynamic Feedback Protocol (DFP) to calculate GPRS/UMTS load-balancing weights dynamically.

With IOS SLB DFP support, a *DFP manager* in a load-balancing environment can initiate a TCP connection with a *DFP agent*. Thereafter, the DFP agent collects status information from one or more real host servers, converts the information to relative weights, and reports the weights to the DFP manager. The DFP manager factors in the weights when load balancing the real servers. In addition to reporting at user-defined intervals, the DFP agent sends an early report if there is a sudden change in a real server's status.

The weights calculated by DFP override the static weights you define using the **weight (server farm)** command. If DFP is removed from the network, IOS SLB reverts to the static weights.

You can define IOS SLB as a DFP manager, as a DFP agent for another DFP manager (such as DistributedDirector), or as both at the same time. In such a configuration, IOS SLB sends periodic reports to DistributedDirector, which uses the information to choose the best server farm for each new connection request. IOS SLB then uses the same information to choose the best real server within the chosen server farm.

DFP also supports the use of multiple DFP agents from different client subsystems (such as IOS SLB and GPRS/UMTS) at the same time.

In GTP load balancing, you can define IOS SLB as a DFP manager and define a DFP agent on each GGSN in the server farm, and the DFP agent can report the weights of the GGSNs. The DFP agents calculate the weight of each GGSN based on CPU utilization, processor memory, and the maximum number of PDP contexts (mobile sessions) that can be activated for each GGSN.

The weight for each GGSN is primarily based on the ratio of existing PDP contexts on the GGSN and the maximum number of allowed PDP contexts. CPU and memory utilization become part of the weight calculation only after the utilization exceeds 85%. Because the maximum number of allowed PDP contexts is considered to be the GGSNs maximum load, you should carefully consider the value that you configure in the **gprs maximum-pdp-context-allowed** command, which defaults to 10000 PDP contexts.

Restrictions

The following restrictions apply when configuring GTP load balancing:

- For GTP load balancing without GTP cause code inspection enabled:
  - Operates in either dispatched mode or directed server NAT mode only.
  - Cannot load balance network-initiated PDP context requests.
  - Does not support the following Cisco IOS SLB functions:
    - Bind IDs
    - Client-assigned load balancing
    - Slow Start
    - Stateful backup
    - Sticky connections
    - Weighted least connections load-balancing algorithm.
- For GTP load balancing *with* GTP cause code inspection enabled:
  - Operates in directed server NAT mode only.
  - Cannot load-balance network-initiated PDP context requests.
  - Requires either the SGSN or the GGSN to echo its peer.
  - Inbound and outbound traffic should be routed via IOS SLB.
  - Does not support the following IOS SLB functions:
    - Bind IDs
    - Client-assigned load balancing
    - Slow Start
    - Sticky connections

Configuring GTP Load Balancing

This section includes the following topics:

GTP Load Balancing Configuration Task List

This section lists the tasks used to configure GTP load balancing. Detailed configuration information is contained in the referenced sections of this or other documents. Required and optional tasks are indicated.

- On the IOS SLB, complete the following tasks:
  - Configuring a Server Farm and Real Server, page 12-8 (Required)
  - Configuring a Virtual Server, page 12-10 (Required)
  - Configuring the virtual IP address as a loopback on each of the GGSNs in the server (Required if using dispatched mode)

    This step is required only if you are using dispatched mode *without* GTP cause code inspection enabled. See the "Configuring a Loopback Interface" section in the *Cisco IOS Interface Configuration Guide* for more information.
  - Configuring a GSN Idle Timer, page 12-12 (Optional if GTP cause code inspection is enabled.)
  - Configuring DFP, page 12-13 (Optional but recommended)
- On the GGSN, complete the following tasks:
  - If using DFP:

    - Configuring the Maximum DFP Weight for a GGSN, page 12-13 (Optional)

    - Configuring the Maximum Number of PDP Contexts for a GGSN, page 12-13 (Optional)
  - Identifying the GGSN Virtual Server to CEF, page 12-14 (Required if using CEF)
  - Routing each GGSN to each associated SGSN (Required)

    The route can be static or dynamic but the GGSN needs to be able to reach the SGSN. For more information, see the "Configuring a Route to the SGSN" section on page 6-3 of the "Configuring Network Access to the GGSN" chapter.
- On the SGSN, route each SGSN to the virtual templates on each associated GGSN, and to the GGSN load-balancing virtual server (Required)

  See the configuration guide for your SGSN for more details.

Configuration Guidelines

When configuring the network shared by IOS SLB and the GGSNs, keep the following considerations in mind:

- Specify static routes (using **ip route** commands) and real server IP addresses (using **real** commands) such that the Layer 2 information is correct and unambiguous.
- Choose subnets carefully, using one of the following methods:
  - Do not overlap virtual template address subnets.
  - Specify next hop addresses to real servers, not to interfaces on those servers.
- IOS SLB supports two types of GTP load balancing:
  - GTP Load Balancing without GTP Cause Code Inspection, page 12-3
  - GTP Load Balancing with GTP Cause Code Inspection, page 12-3
- IOS SLB supports both GTP v0 and GTP v1. Support for GTP enables IOS SLB to become "GTP aware," extending IOS SLB's knowledge into Layer 5.
- If you have enabled Cisco Express Forwarding (CEF) on a GGSN, you must identify the IP address of the GGSN virtual server to CEF. If you have *not* enabled CEF on the GGSN, do not perform this task. See the "Identifying the GGSN Virtual Server to CEF" section on page 12-14 for more details.

Configuring a Server Farm and Real Server

When you configure the server farm and real server on the IOS SLB for GTP load balancing, use the following guidelines to ensure proper configuration:

- If GTP cause code inspection is not enabled, accept the default setting (the weighted round robin algorithm) for the **predictor** command.

  If GTP cause code inspection is enabled, you can specify either the weighted round robin algorithm (**roundrobin**) or the weighted least connections (**leastconns**) algorithm.
- Specify the IP addresses (virtual template addresses, for Cisco GGSNs) of the real servers performing the GGSN function, using the **real** command.
- Specify a reassign threshold less than the SGSN's N3-REQUESTS counter value using the **reassign** command.

To configure an IOS SLB server farm, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip slb serverfarm serverfarm-name` `Router(config-slb-sfarm)#`	Adds a server farm definition to the IOS SLB configuration and enters server farm configuration mode.
Step 2	`Router(config-slb-sfarm)# predictor [roundrobin | leastconns]`	Specifies the algorithm to be used to determine how a real server is selected. **Note** In GTP load balancing without GTP cause code inspection enabled, you must accept the default setting (the weighted round robin algorithm). See the following sections for more details about each algorithm: • Weighted Round Robin, page 12-4 • Weighted Least Connections, page 12-4
Step 3	`Router(config-slb-sfarm)# nat server`	(Required if GTP cause code inspection is enabled) Configures NAT server address translation mode on the server farm.
Step 4	`Router(config-slb-sfarm)# real ip-address [port]`	Identifies a real GGSN as a member of a server farm using the IP address of the GGSN's virtual template interface, and enters real server configuration mode.
Step 5	`Router(config-slb-real)# faildetect numconns number-conns [numclients number-clients]`	(Optional) Specifies the number of consecutive connection failures and, optionally, the number of unique client connection failures, that constitute failure of the real server. In GTP load balancing, if there is only one SGSN in your environment, specify the **numclients** keyword with a value of 1.
Step 6	`Router(config-slb-real)# maxconns number-conns`	(Optional) Specifies the maximum number of active connections allowed on the real server at one time. **Note** The impact of this command in GTP load balancing *without* GTP cause code inspection enabled is minimal because sessions are very short-lived.

	Command	Purpose
Step 7	`Router(config-slb-real)# reassign threshold`	(Optional) Specifies the threshold of consecutive unacknowledged synchronizations or create PDP context requests that, if exceeded, result in an attempted connection to a different real server. **Note** In GTP load balancing, you must specify a reassign threshold less than the SGSN's N3-REQUESTS counter value.
Step 8	`Router(config-slb-real)# retry retry-value`	(Optional) Specifies the interval, in seconds, to wait between the detection of a server failure and the next attempt to connect to the failed server.
Step 9	`Router(config-slb-real)# weight weighting-value`	(Optional) Specifies the real server's workload capacity relative to other servers in the server farm. **Note** If you use DFP, the static weights you define using the **weight (server farm)** command are overridden by the weights calculated by DFP. If DFP is removed from the network, IOS SLB reverts to the static weights.
Step 10	`Router(config-slb-real)# inservice`	Enables the real server for use by IOS SLB.

Configuring a Virtual Server

When you configure the virtual server on the IOS SLB for GTP load balancing, use the following guidelines to ensure proper configuration:

- Specify a virtual GGSN IP address as the virtual server, and use the **udp** keyword option.
- To load-balance GTP v1 sessions, specify port number **2123**, if the GGSNs and SGSNs are in compliance with the ETSI standard, or specify port number **0** or **any** to configure an all-port virtual server (that is, a virtual server that accepts flows destined for all ports).
- To load-balance GTP v0 sessions, specify port number **3386**, if the GGSNs and SGSNs are in compliance with the ETSI standard, or specify port number **0** or **any** to configure an all-port virtual server.
- To enable GTP load balancing *without* GTP cause code inspection, specify the **service gtp** keyword option.
- To enable GTP load balancing *with* GTP cause code inspection, specify the **service gtp-inspect** keyword option.

  In GTP load balancing *without* GTP cause code inspection enabled, when you configure the idle timer using the **idle** command, specify an idle timer greater than the longest possible interval between PDP context requests on the SGSN.

To configure an IOS SLB virtual server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip slb vserver virtual_server-name`	Identifies a virtual server and enters virtual server configuration mode.

Step 2	`Router(config-slb-vserver)# virtual ip-addr [netmask [group]] {esp | gre | protocol}`
	or
	`Router(config-slb-vserver)# virtual ip-addr [netmask [group]] {tcp | udp} [port | any] [service service]`
		Specifies the virtual server IP address, type of connection, and optional TCP or UDP port number, IKE (ISAKMP) or WSP setting, and service coupling.
		Note For GTP load balancing:
		– Specify a virtual GGSN IP address as the virtual server, and specify the **udp** keyword option.
		– To load-balance GTP v1 sessions, specify port number **2123**, if the GGSNs and SGSNs are in compliance with the ETSI standard, or specify port number **0** or **any** to configure an all-port virtual server (that is, a virtual server that accepts flows destined for all ports).
		– To load-balance GTP v0 sessions, specify port number **3386**, if the GGSNs and SGSNs are in compliance with the ETSI standard, or specify port number **0** or **any** to configure an all-port virtual server.
		– To enable GTP load balancing *without* GTP cause code inspection, specify the **service gtp** keyword option.
		– To enable GTP load balancing *with* GTP cause code inspection, specify the **service gtp-inspect** keyword option.

Step 3	`Router(config-slb-vserver)# serverfarm primary-farm [backup backup-farm]`	Associates a real server farm with a virtual server, or configures a backup server farm.

	Command	Purpose
Step 4	`Router(config-slb-vserver)# idle [gtp request] duration`	(Optional) Specifies the minimum amount of time IOS SLB maintains connection context in the absence of packet activity. **Note** In GTP load balancing *without* GTP cause code inspection enabled, specify an idle timer greater than the longest possible interval between PDP context requests on the SGSN. The **gtp request** keyword option is only supported in GTP load balancing with GTP cause code inspection enabled.
Step 5	`Router(config-slb-vserver)# inservice`	Enables the virtual server for use by IOS SLB.
Step 6	`Router(config-slb-vserver)# client {ip-address network-mask [exclude] | gtp carrier-code [code]}`	(Optional) Specifies which clients are allowed to use the virtual server. **Note** GTP load balancing supports only the **gtp carrier-code** option, and only if GTP cause code inspection is enabled.
Step 7	`Router(config-slb-vserver)# replicate casa listen-ip remote-ip port [interval] [password [0 | 7] password timeout]`	(Optional) Configures a stateful backup of IOS SLB decision tables to a backup switch. **Note** GTP load balancing *without* GTP cause code inspection enabled does not support this command.

Configuring a GSN Idle Timer

When GTP cause code inspection is enabled, you can configure the amount of time the IOS SLB will maintain sessions to and from and idle GGSN or SGSN.

To configure a GSN idle timer, enter the following command in global configuration mode on the IOS SLB:

Command	Purpose
`Router(config)# ip slb timers gtp gsn duration`	Changes the amount of time IOS SLB maintains sessions to and from an idle GGSN or SGSN.

Configuring DFP

You can define IOS SLB as a DFP manager, as a DFP agent for another DFP manager (such as DistributedDirector), or as both at the same time. Depending on your network configuration, you might enter the commands for configuring IOS SLB as a DFP manager and the commands for configuring IOS SLB as a DFP agent on the same device or on different devices.

To configure IOS SLB as a DFP manager, and to identify a DFP agent with which IOS SLB can initiate connections, use the following commands beginning in global configuration mode:

	Command	Description
Step 1	`Router(config)# ip slb dfp [password [0|7] password [timeout]]`	Configures DFP, supplies an optional password, and enters DFP configuration mode.
Step 2	`Router(config-slb-dfp)# agent ip_address port-number [timeout [retry_count [retry_interval]]]`	Identifies a DFP agent to which IOS SLB can connect.

To configure IOS SLB as a DFP agent, see the *DFP Agent Subsystem* feature module.

Configuring the Maximum DFP Weight for a GGSN

If you use DFP with GTP load balancing, each GGSN that acts as a DFP agent has a maximum weight that it can send to a DFP manager. For each GGSN, you can accept the default maximum weight, or you can specify a different maximum weight.

To specify the maximum weight for a GGSN, use the following command in global configuration mode on the GGSN:

Command	Purpose
`Router(config)# gprs dfp max-weight [max-weight-value]`	Specifies the maximum weight of a GGSN that is acting as a DFP agent.

Configuring the Maximum Number of PDP Contexts for a GGSN

If you use DFP with GTP load balancing, you must specify a maximum number of PDP contexts for each GGSN, using the **gprs maximum-pdp-context-allowed** command. *Do not* accept the default value of 10000 PDP contexts. A value of **45000** is recommended. Significantly lower values, including the default value of 10000, can impact performance in a GPRS/UMTS load-balancing environment.

To configure a maximum number of PDP contexts for a GGSN, use the following command in global configuration mode on the GGSN:

Command	Purpose
`Router(config)# gprs maximum-pdp-context-allowed [pdp-contexts]`	Specifies the maximum number of PDP contexts (mobile sessions) that can be activated on the GGSN.

For more information about the number of PDP contexts supported on a GGSN, see the "Configuring the Maximum Number of PDP Contexts on the GGSN" section on page 4-15 of the "Configuring GGSN GTP Services" chapter.

Identifying the GGSN Virtual Server to CEF

If you have enabled CEF on a GGSN and are using dispatched mode, you must identify the IP address of the GGSN virtual server to CEF. (This IP address is also a loopback address on the GGSN.)

If you have *not* enabled CEF on the GGSN and are not using dispatched mode, do not perform this task.

To identify the IP address of the GGSN virtual server to CEF, use the following command in global configuration mode on the GGSN:

Command	Purpose
`Router(config)# gprs slb cef [virtual-server-address]`	Specifies the IP address of the GGSN virtual server instance used by clients to connect to the server farm, for use by CEF. This command is required only if the GGSN is using CEF switching. Do not use this command if the GGSN is not using CEF switching.

Verifying the IOS SLB Configuration

This section describes how to verify the IOS SLB configuration. It includes the following topics:

- Verifying the Virtual Server, page 12-14
- Verifying the Server Farm, page 12-15
- Verifying the Clients, page 12-15
- Verifying IOS SLB Connectivity, page 12-15

Verifying the Virtual Server

The following **show ip slb vserver** command verifies the configuration of the virtual servers PUBLIC_HTTP and RESTRICTED_HTTP:

```
Router# show ip slb vserver

slb vserver      prot  virtual               state         conns
-------------------------------------------------------------------
PUBLIC_HTTP      TCP   10.0.0.1:80           OPERATIONAL     0
RESTRICTED_HTTP  TCP   10.0.0.2:80           OPERATIONAL     0
Router#
```

Verifying the Server Farm

The following **show ip slb reals** command displays the status of server farms PUBLIC and RESTRICTED, the associated real servers, and their status:

```
Router# show ip slb real

real                    farm name        weight   state          conns
-----------------------------------------------------------------------
10.1.1.1                 PUBLIC           8       OPERATIONAL      0
10.1.1.2                 PUBLIC           8       OPERATIONAL      0
10.1.1.3                 PUBLIC           8       OPERATIONAL      0
10.1.1.20                RESTRICTED       8       OPERATIONAL      0
10.1.1.21                RESTRICTED       8       OPERATIONAL      0
Router#
```

The following **show ip slb serverfarm** command displays the configuration and status of server farms PUBLIC and RESTRICTED:

```
Router# show ip slb serverfarm

server farm      predictor    nat   reals   bind id
--------------------------------------------------
PUBLIC           ROUNDROBIN   none  3       0
RESTRICTED       ROUNDROBIN   none  2       0
Router#
```

Verifying the Clients

The following **show ip slb conns** command verifies the restricted client access and status:

```
Router# show ip slb conns

vserver         prot client                real                  state     nat
-------------------------------------------------------------------------------
RESTRICTED_HTTP TCP  10.4.4.0:80           10.1.1.20             CLOSING   none
Router#
```

The following **show ip slb conns** command displays detailed information about the restricted client access status:

```
Router# show ip slb conns client 10.4.4.0 detail
VSTEST_UDP, client = 10.4.4.0:80
  state = CLOSING, real = 10.1.1.20, nat = none
  v_ip = 10.0.0.2:80, TCP, service = NONE
  client_syns = 0, sticky = FALSE, flows attached = 0
Router#
```

Verifying IOS SLB Connectivity

To verify that the IOS SLB feature has been installed and is operating correctly, ping the real servers from the IOS SLB switch, then ping the virtual servers from the clients.

The following **show ip slb stats** command displays detailed information about the IOS SLB network status:

```
Router# show ip slb stats
Pkts via normal switching:    0
Pkts via special switching:   0
Pkts via slb routing:         0
Pkts Dropped:                 0
```

```
Connections Created:          0
Connections Established:      0
Connections Destroyed:        0
Connections Reassigned:       0
Zombie Count:                 0
Connections Reused:           0
Connection Flowcache Purges:  0
Failed Connection Allocs:     0
Failed Real Assignments:      0
RADIUS framed-ip Sticky Count:0
RADIUS username Sticky Count: 0
```

See the "Monitoring and Maintaining the IOS SLB Feature" section on page 12-16 for additional commands used to verify IOS SLB networks and connections.

Monitoring and Maintaining the IOS SLB Feature

To obtain and display runtime information about IOS SLB for the GGSN, use the following commands in privileged EXEC mode:

Command	Purpose
Router# **show ip slb conns** [**vserver** *virtual_server-name* | **client** *ip-address* | **firewall** *firewallfarm-name*] [**detail**]	Displays all connections handled by IOS SLB, or, optionally, only those connections associated with a particular virtual server or client.
Router# **show ip slb dfp** [**agent** *agent_ip_address port-number* | **manager** *manager_ip_address* | **detail** | **weights**]	Displays information about DFP and DFP agents, and about the weights assigned to real servers.
Router# **show ip slb gtp** {**gsn** [*gsn-ip-address*] | **nsapi** [*nsapi-key*] [**detail**]	Displays IOS SLB GTP information.
Router# **show ip slb reals** [**sfarm** *server-farm*] [**detail**]	Displays information about the real servers defined to IOS SLB.
Router# **show ip slb replicate**	Displays information about the IOS SLB replication configuration.
Router# **show ip slb serverfarms** [**name** *serverfarm-name*] [**detail**]	Displays information about the server farms defined to IOS SLB.
Router# **show ip slb sessions** [**gtp** | **gtp-inspect** | **radius**] [**vserver** *virtual-server*] [**client** *ip-addr netmask*] [**detail**]	Displays information about sessions handled by IOS SLB.
Router# **show ip slb stats**	Displays IOS SLB statistics.
Router# **show ip slb vserver** [**name** *virtual_server*] [**redirect**] [**detail**]	Displays information about the virtual servers defined to IOS SLB.

Configuration Examples

This section provides real-world examples of the GGSN IOS SLB examples. For a complete description of the GGSN commands in this section, see the Cisco IOS Mobile Wireless Command Reference. For a complete description of the IOS SLB commands in this section, see the *IOS Server Load Balancing*, 12.1(13)E3 documentation.

This section includes the following examples:

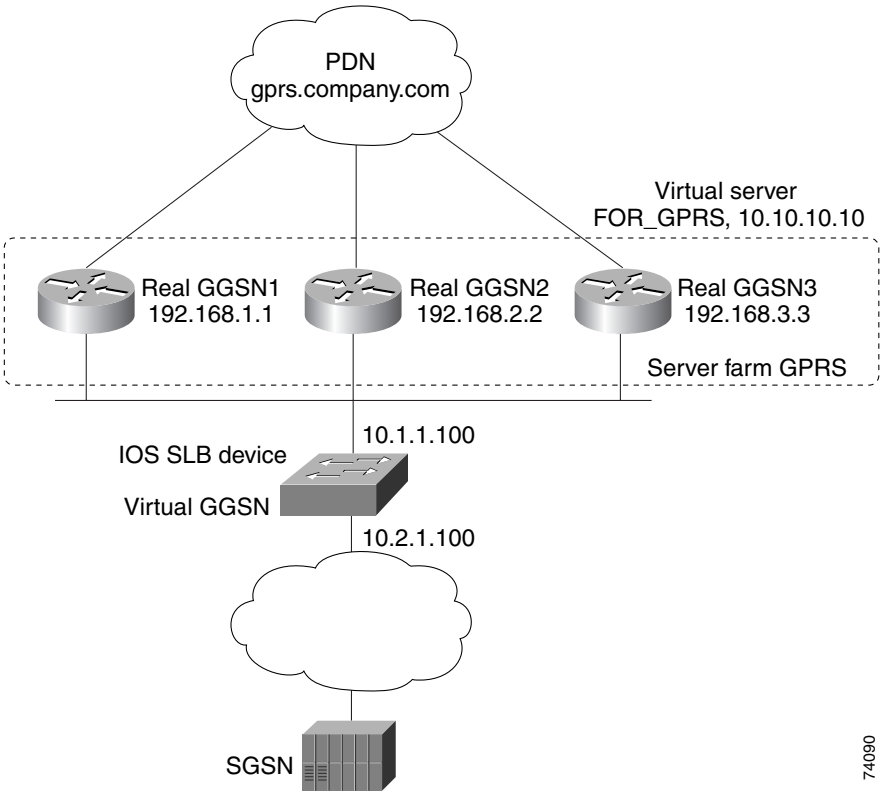
- IOS SLB with GTP Load Balancing Configuration Example, page 12-17
- IOS SLB with GTP Load Balancing and NAT Example, page 12-22
- IOS SLB with GTP Load Balancing, NAT, and GTP Cause Code Inspection Example, page 12-25

IOS SLB with GTP Load Balancing Configuration Example

Figure 2 shows a typical GTP load-balancing configuration *without* GTP cause code inspection enabled. In this configuration:

- IOS SLB can balance GTP flows across multiple real GGSNs. The SGSN “sees” the real GGSNs as a single virtual GGSN. This configuration increases the flow-handling capability of the real GGSNs and increases the reliability and availability.
- The ip address of the SGSN is 10.111.111.111.
- The virtual template address of GGSN1 is 192.168.1.1.
- The virtual template address of GGSN2 is 192.168.2.2.
- The virtual template address of GGSN3 is 192.168.3.3.

Figure 2 IOS SLB with GTP Load Balancing



Following are the configuration statements for the configuration shown in Figure 2:

- IOS SLB Configuration Statements, page 12-19
- GGSN1 Configuration Statements, page 12-19
- GGSN2 Configuration Statements, page 12-20
- GGSN3 Configuration Statements, page 12-21

For more detailed GGSN configuration examples, see the *Cisco IOS Mobile Wireless Configuration Guide*.

IOS SLB Configuration Statements

```
hostname GTP_SLB
!
ip domain-name gprs.com
!
ip slb serverfarm GPRS
 real 192.168.1.1
  weight 1
  faildetect numconns 1 numclients 1
  inservice
!
 real 192.168.2.2
  weight 1
  faildetect numconns 1 numclients 1
  inservice
!
 real 192.168.3.3
  weight 1
  faildetect numconns 1 numclients 1
  inservice
!
ip slb vserver FOR_GPRS
 virtual 10.10.10.10 udp 3386 service gtp
 serverfarm GPRS
 inservice
!
ip slb dfp password Cookies 0
 agent 10.1.1.201 1111 30 0 10
 agent 10.1.1.202 1111 30 0 10
 agent 10.1.1.203 1111 30 0 10
!
interface FastEthernet1/0
 description TO SERVERFARM GPRS
 ip address 10.1.1.100 255.255.255.0
 no ip redirects
 duplex half
!
interface FastEthernet3/0
 description TO SGSN
 ip address 10.2.1.100 255.255.255.0
 no ip mroute-cache
 duplex half
!
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
ip route 192.168.1.1 255.255.255.255 10.1.1.201
ip route 192.168.2.2 255.255.255.255 10.1.1.202
ip route 192.168.3.3 255.255.255.255 10.1.1.203
```

GGSN1 Configuration Statements

```
service gprs ggsn
!
hostname GGSN1
!
ip dfp agent gprs
 port 1111
 password Cookies 0
 inservice
!
ip domain-name gprs.com
!
```

```
interface loopback 1
 description LOOPBACK SAME AS IOS SLB VSERVER ADDRESS
 ip address 10.10.10.10 255.255.255.255
 no ip route-cache
 no ip mroute-cache
!
interface FastEthernet1/0
 description TO SLB
 ip address 10.1.1.201 255.255.255.0
 ip directed-broadcast
 no ip mroute-cache
 duplex half
!
interface Virtual-Template1
 description GTP VIRTUAL TEMPLATE
 ip address 192.168.1.1 255.255.255.0
 encapsulation gtp
 gprs access-point-list gprs1
!
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
!
gprs access-point-list gprs1
  access-point 1
   access-point-name gprs.company.com
   access-mode non-transparent
   ip-address-pool dhcp-proxy-client
   dhcp-server 10.100.0.5 10.100.0.6
   dhcp-gateway-address 10.27.3.1
   exit
!
gprs maximum-pdp-context-allowed 45000
gprs qos map canonical-qos
gprs gtp path-echo-interval 0
gprs dfp max-weight 32
gprs slb cef 10.10.10.10
```

GGSN2 Configuration Statements

```
service gprs ggsn
!
hostname GGSN2
!
ip dfp agent gprs
 port 1111
 password Cookies 0
 inservice
!
ip domain-name gprs.com
!
interface loopback 1
 description LOOPBACK SAME AS IOS SLB VSERVER ADDRESS
 ip address 10.10.10.10 255.255.255.255
 no ip route-cache
 no ip mroute-cache
!
interface FastEthernet1/0
 description TO SLB
 ip address 10.1.1.202 255.255.255.0
 ip directed-broadcast
 no ip mroute-cache
 duplex half
!
```

```
interface Virtual-Template1
 description GTP VIRTUAL TEMPLATE
 ip address 192.168.2.2 255.255.255.0
 encapsulation gtp
 gprs access-point-list gprs1
!
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
!
gprs access-point-list gprs1
  access-point 1
   access-point-name gprs.company.com
   access-mode non-transparent
   ip-address-pool dhcp-proxy-client
   dhcp-server 10.100.0.5 10.100.0.6
   dhcp-gateway-address 10.27.3.1
   exit
!
gprs maximum-pdp-context-allowed 45000
gprs qos map canonical-qos
gprs gtp path-echo-interval 0
gprs dfp max-weight 32
gprs slb cef 10.10.10.10
```

GGSN3 Configuration Statements

```
service gprs ggsn
!
hostname GGSN3
!
ip dfp agent gprs
 port 1111
 password Cookies 0
 inservice
!
ip domain-name gprs.com
!
interface loopback 1
 description LOOPBACK SAME AS IOS SLB VSERVER ADDRESS
 ip address 10.10.10.10 255.255.255.255
 no ip route-cache
 no ip mroute-cache
!
interface FastEthernet1/0
 description TO SLB
 ip address 10.1.1.203 255.255.255.0
 ip directed-broadcast
 no ip mroute-cache
 duplex half
!
interface Virtual-Template1
 description GTP VIRTUAL TEMPLATE
 ip address 192.168.3.3 255.255.255.0
 encapsulation gtp
 gprs access-point-list gprs1
!
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
!

gprs access-point-list gprs1
  access-point 1
   access-point-name gprs.company.com
   access-mode non-transparent
```

```
   ip-address-pool dhcp-proxy-client
   dhcp-server 10.100.0.5 10.100.0.6
   dhcp-gateway-address 10.27.3.1
   exit
!
gprs maximum-pdp-context-allowed 45000
gprs qos map canonical-qos
gprs gtp path-echo-interval 0
gprs dfp max-weight 32
gprs slb cef 10.10.10.10
```

IOS SLB with GTP Load Balancing and NAT Example

The following example uses the same basic configuration as in the "IOS SLB with GTP Load Balancing Configuration Example" section on page 12-17, including the network shown in Figure 2, but with the addition of NAT:

- IOS SLB Configuration Statements, page 12-22
- GGSN1 Configuration Statements, page 12-23
- GGSN2 Configuration Statements, page 12-24
- GGSN3 Configuration Statements, page 12-24

IOS SLB Configuration Statements

```
hostname GTP_SLB
!
ip domain-name gprs.com
!
ip slb serverfarm GPRS
 nat server
 real 192.168.1.1
  weight 1
  faildetect numconns 1 numclients 1
  inservice
!
 real 192.168.2.2
  weight 1
  faildetect numconns 1 numclients 1
  inservice
!
 real 192.168.3.3
  weight 1
  faildetect numconns 1 numclients 1
  inservice
!
ip slb vserver FOR_GPRS
 virtual 10.10.10.10 udp 3386 service gtp
 serverfarm GPRS
 inservice
!

ip slb dfp password Cookies 0
 agent 10.1.1.201 1111 30 0 10
 agent 10.1.1.202 1111 30 0 10
 agent 10.1.1.203 1111 30 0 10
!
interface FastEthernet1/0
 description TO SERVERFARM GPRS
```

```
 ip address 10.1.1.100 255.255.255.0
 no ip redirects
 duplex half
!
interface FastEthernet3/0
 description TO SGSN
 ip address 10.2.1.100 255.255.255.0
 no ip mroute-cache
 duplex half
!
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
ip route 192.168.1.1 255.255.255.255 10.1.1.201
ip route 192.168.2.2 255.255.255.255 10.1.1.202
ip route 192.168.3.3 255.255.255.255 10.1.1.203
```

GGSN1 Configuration Statements

```
service gprs ggsn
!
hostname GGSN1
!
ip dfp agent gprs
 port 1111
 password Cookies 0
 inservice
!
ip domain-name gprs.com
!
interface FastEthernet1/0
 description TO SLB
 ip address 10.1.1.201 255.255.255.0
 ip directed-broadcast
 no ip mroute-cache
 duplex half
!
interface Virtual-Template1
 description GTP VIRTUAL TEMPLATE
 ip address 192.168.1.1 255.255.255.0
 encapsulation gtp
 gprs access-point-list gprs1
!
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
!
gprs access-point-list gprs1
  access-point 1
   access-point-name gprs.company.com
   access-mode non-transparent
   ip-address-pool dhcp-proxy-client
   dhcp-server 10.100.0.5 10.100.0.6
   dhcp-gateway-address 10.27.3.1
   exit
!



gprs maximum-pdp-context-allowed 45000
gprs qos map canonical-qos
gprs gtp path-echo-interval 0
gprs dfp max-weight 32
```

GGSN2 Configuration Statements

```
service gprs ggsn
!
hostname GGSN2
!
ip dfp agent gprs
 port 1111
 password Cookies 0
 inservice
!
ip domain-name gprs.com
!
interface FastEthernet1/0
 description TO SLB
 ip address 10.1.1.202 255.255.255.0
 ip directed-broadcast
 no ip mroute-cache
 duplex half
interface Virtual-Template1
 description GTP VIRTUAL TEMPLATE
 ip address 192.168.2.2 255.255.255.0
 encapsulation gtp
 gprs access-point-list gprs1
!
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
!
gprs access-point-list gprs1
  access-point 1
   access-point-name gprs.company.com
   access-mode non-transparent
   ip-address-pool dhcp-proxy-client
   dhcp-server 10.100.0.5 10.100.0.6
   dhcp-gateway-address 10.27.3.1
   exit
!
gprs maximum-pdp-context-allowed 45000
gprs qos map canonical-qos
gprs gtp path-echo-interval 0
gprs dfp max-weight 32
```

GGSN3 Configuration Statements

```
service gprs ggsn
!
hostname GGSN3
!
ip dfp agent gprs
 port 1111
 password Cookies 0
 inservice
!
ip domain-name gprs.com
!
interface FastEthernet1/0
 description TO SLB
 ip address 10.1.1.203 255.255.255.0
 ip directed-broadcast
 no ip mroute-cache
 duplex half
!
```

```
interface Virtual-Template1
 description GTP VIRTUAL TEMPLATE
 ip address 192.168.3.3 255.255.255.0
 encapsulation gtp
 gprs access-point-list gprs1
!
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
!
gprs access-point-list gprs1
  access-point 1
   access-point-name gprs.company.com
   access-mode non-transparent
   ip-address-pool dhcp-proxy-client
   dhcp-server 10.100.0.5 10.100.0.6
   dhcp-gateway-address 10.27.3.1
   exit
!
gprs maximum-pdp-context-allowed 45000
gprs qos map canonical-qos
gprs gtp path-echo-interval 0
gprs dfp max-weight 32
```

IOS SLB with GTP Load Balancing, NAT, and GTP Cause Code Inspection Example

The following example uses the same basic configuration as in the "IOS SLB with GTP Load Balancing and NAT Example" section on page 12-22, including the network shown in Figure 2, but with the GTP cause code inspection enabled. In this configuration:

- The GSN idle timer is set to 20 seconds.
- The GTP request idle timer is set to 15 seconds.
- The virtual server accepts PDP context creates only from International Mobile Subscriber IDs (IMSIs) with carrier code **mcc 222 mnc 22**.

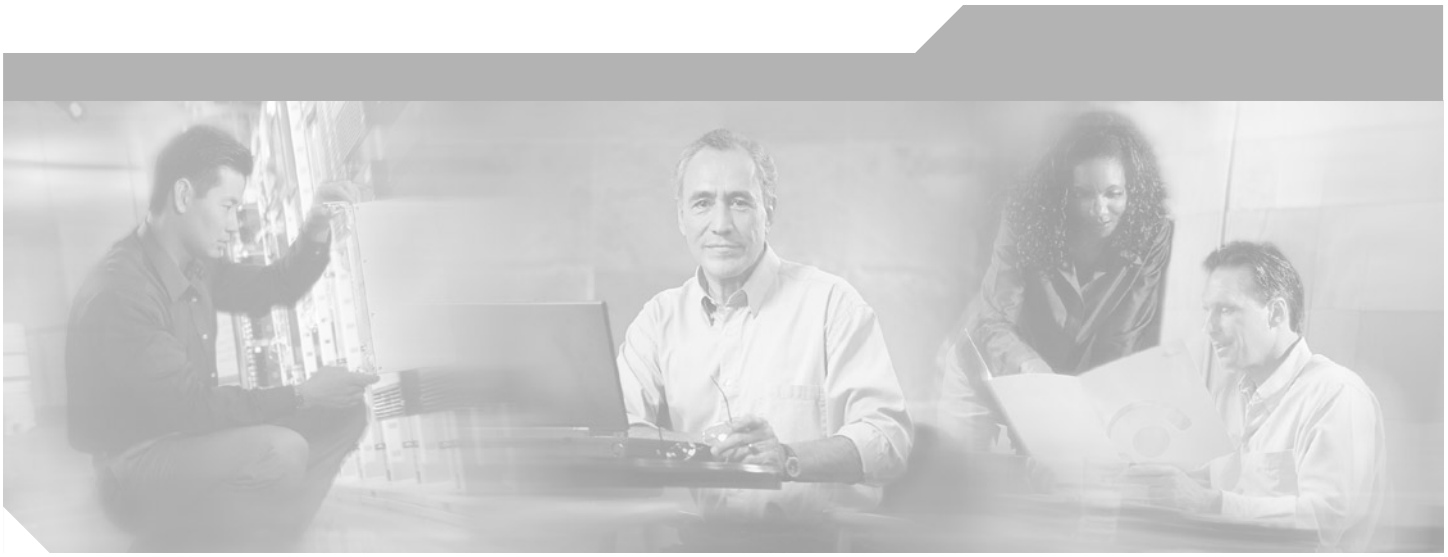
Following are the configuration statements for the configuration shown in Figure 2, with the addition of NAT and GTP cause code inspection support:

- IOS SLB Configuration Statements, page 12-26
- GGSN1 Configuration Statements, page 12-23 (no change for GTP cause code inspection)
- GGSN2 Configuration Statements, page 12-24 (no change for GTP cause code inspection)
- GGSN3 Configuration Statements, page 12-24 (no change for GTP cause code inspection)

IOS SLB Configuration Statements

```
hostname GTP_SLB
!
ip domain-name gprs.com
!
ip slb timers gtp gsn 20
!
ip slb serverfarm GPRS
 nat server
 real 192.168.1.1
  weight 1
  faildetect numconns 1 numclients 1
  inservice
!
 real 192.168.2.2
  weight 1
  faildetect numconns 1 numclients 1
  inservice
!
 real 192.168.3.3
  weight 1
  faildetect numconns 1 numclients 1
  inservice
!
ip slb vserver FOR_GPRS
 virtual 10.10.10.10 udp 0 service gtp-inspect
 idle gtp request 15
 client gtp carrier-code mcc 222 mnc 22
 serverfarm GPRS
 inservice
!
ip slb dfp password Cookies 0
 agent 10.1.1.201 1111 30 0 10
 agent 10.1.1.202 1111 30 0 10
 agent 10.1.1.203 1111 30 0 10
!
interface FastEthernet1/0
 description TO SERVERFARM GPRS
 ip address 10.1.1.100 255.255.255.0
 no ip redirects
 duplex half
!
interface FastEthernet3/0
 description TO SGSN
 ip address 10.2.1.100 255.255.255.0
 no ip mroute-cache
 duplex half
!
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
ip route 192.168.1.1 255.255.255.255 10.1.1.201
ip route 192.168.2.2 255.255.255.255 10.1.1.202
ip route 192.168.3.3 255.255.255.255 10.1.1.203
```

CISCO SYSTEMS



PART 2

GTP Director Module
Release 1.0

Overview of GDM

This chapter provides a brief introduction to the GTP Director Module (GDM) and its implementation in the Cisco IOS software.

This chapter includes the following sections:

Feature Description

GGSN Release 3.0 and later adds GDM as part of the GGSN feature set in the Cisco IOS software. GDM extends some of the benefits that are available on a Cisco Systems GGSN, to GPRS/UMTS environments where non-Cisco GGSNs are implemented.

These benefits include reducing APN provisioning requirements in the GPRS/UMTS PLMN, while also providing simple, round-robin load balancing for the GGSNs. A network using GDM has the added benefit of the Cisco Systems Hot Standby Router Protocol (HSRP) to support increased network availability using a backup GDM router.

Like the Cisco Systems' GGSN, GDM provides access to multiple destination networks through a virtual APN that is provisioned at the Home Location Register (HLR). GDM's virtual APN support simplifies the maintenance and provisioning issues in the GPRS/UMTS PLMN significantly. In this one-to-many model, one APN can be provisioned for multiple subscribers, and that one APN can provide access to many real destination networks. By implementing virtual APN support, service providers can add new access points without having to provision the HLR.

Using DNS, GDM also provides round-robin load balancing for those GGSNs that support access to the same destination networks.

To provide increased network availability, a backup GDM router can be configured to automatically switch over and become the primary GDM router using HSRP. The backup GDM router can provide access to the GGSNs if the primary GDM router, or even a critical interface on the primary GDM router, becomes unavailable.

Although GDM is part of the GGSN feature set, it cannot coexist on a router that is also configured as a Cisco Systems GGSN. However, GDM can be used in a mixed environment of Cisco and non-Cisco GGSNs.

GDM does not add any value to an environment that includes only Cisco Systems GGSNs, considering that Cisco Systems GGSNs have alternative and enhanced load balancing solutions, and can natively provide virtual APN support. For more information about load balancing options for a Cisco Systems GGSN, see the "Configuring Load Balancing on the GGSN" chapter. For information about virtual APN support, see the "Configuring Virtual APN Access on the GGSN" section on page 6-29.

Request Processing by GDM

This section describes how GDM processes create PDP context requests and retries of those requests, and describes several different request processing scenarios. This section includes the following topics:

Overview of Request Processing by GDM

GDM's role in the GPRS/UMTS PLMN is to facilitate the processing of create PDP context requests between an SGSN and one or more GGSNs. GDM processes create PDP context requests sent by an SGSN, and forwards them to the appropriate destination GGSN. GDM does not monitor whether or not a create PDP context request has been successful, or if a path has been established between an SGSN and GGSN for a particular tunnel ID (TID).

In the case of an unsuccessful session establishment, GDM continues to receive retry requests from an SGSN. GDM processes the retries of a create PDP context request for a particular TID and forwards those retries to the GGSN to which the original request was sent. However, GDM only processes those retries for a configurable period of time. GDM forwards retries of a create PDP context request to a GGSN for 30 seconds (default), or for the amount of time that you have configured in the **gprs gtp-director retry-timeout** command.

Once GDM has sent create PDP context requests to a GGSN and has processed any retries, GDM is no longer involved in any other forms of request processing for that PDP context. User authentication for a PDP context request is handled as usual between the GGSN and the authentication, authorization, and accounting (AAA) server.

All of the other signaling request processing occurs directly between the SGSN and the GGSN, over the GTP path established between them. GDM is never part of the GTP path, and data does not flow through GDM. The GTP path remains as usual between the SGSN and a GGSN for a PDP context. For troubleshooting purposes, it is important to note that GDM is never even aware of whether a PDP context has been successfully established with a GGSN.

GDM is not involved in the processing of the following types of requests:

- Echo Requests
- Delete PDP Context Requests
- Update Requests

Request Processing Using a Virtual APN

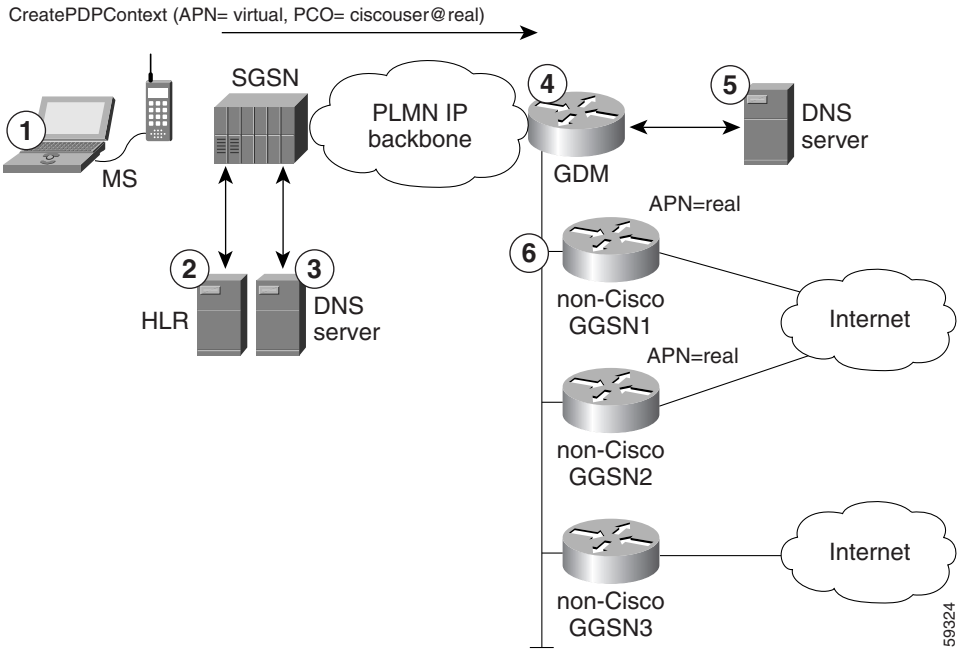
In the GDM environment using virtual APN support, a virtual APN is used to select the Cisco Systems GDM router. GDM facilitates the processing of the create PDP context request to the real APNs through the appropriate GGSNs. The GGSNs always provide the physical connectivity to the real target network.

To implement virtual APN support using GDM, you need to determine the name(s) of the virtual access point(s) that you want subscribers to use for access to one or more real APNs that are configured on your GGSNs.

Figure 13-1 shows how GDM supports a create PDP context request from an MS processed through a virtual APN using GDM in its intended router environment where non-Cisco GGSNs are in use.

Note Recall that you can also use GDM in a mixed environment of Cisco and non-Cisco GGSNs, or in an all Cisco GGSN environment. However, for an environment using only Cisco GGSNs, there are alternative and enhanced load balancing solutions and virtual APN support is already available, which makes GDM less worthwhile for that environment.

Figure 13-1 Virtual APN PDP Context Activation Using GDM and non-Cisco GGSNs



1.	At the MS, the subscriber connects to the network with a username in the form of login@domain, such as *ciscouser@real* and specifies the name of the virtual APN.
2.	The SGSN contacts the HLR to acquire subscription information for the MS. The HLR has a list of the APNs for which the MS is a valid subscriber. The HLR returns to the SGSN the list of subscribed APNs for the MS. The SGSN verifies that the APN called *virtual* is part of the user's subscribed list of APNs.

3. After the SGSN verifies that the virtual APN is valid for the MS user, the SGSN performs a DNS query for the APN named *virtual*, and DNS returns the IP address of GDM.

   The SGSN sends a create PDP context request to GDM using the APN of *virtual*. The create PDP context also includes the username and password in the form of login@domain (which is *ciscouser@real* in our example) in the protocol configuration option (PCO) information element (IE).

4. GDM reads the create PDP context request and extracts the APN of *virtual* from the APN IE. GDM extracts the tunnel ID (TID) and verifies whether it already has a pending PDP context request for that TID. GDM extracts the username from the PCO IE.

5. If no pending request is found, GDM performs a DNS query for the domain that was extracted from the PCO (in this case, *real*). The domain from the information in the PCO, corresponds to the real target network on the GGSN.

   The DNS server returns up to 8 addresses of the GGSNs that can provide connectivity to the real target network to GDM. When multiple addresses are returned by DNS, GDM uses the first address that is provided in the list.

6. GDM forwards a create PDP context request from the SGSN to the target GGSN. In the APN IE of the request, GDM replaces the original APN of *virtual* with the domain name found of *real*. The username field now contains only the username, which is *ciscouser* in this example.

7. The real GGSN processes the create PDP context request using the real APN of *real*, which corresponds to the original domain name requested by the MS. The address of GDM is replaced by the address of the GGSN, and the response to the create PDP context request is sent by the real GGSN to the SGSN. GDM is no longer involved in the processing of the PDP context.

8. If successful, the SGSN sends an activate PDP context message to the MS and the GTP path is established between the SGSN and the GGSN.

   If unsuccessful, the SGSN sends a retry of the create PDP context request to GDM. If the retry timeout period for that TID has not expired, GDM forwards the create PDP context request to the GGSN to which the original request was sent. GDM continues to process any retry requests it receives for that TID until the retry timeout period is reached.

Request Processing Scenarios

GDM processes a PDP context according to the content found in the APN Information Element (IE) and the Protocol Configuration Option (PCO) of the create PDP context request, according to the following scenarios:

- **CreatePDPContext (APN=virtual, PCO=ciscouser@real)**

  In this format the APN IE exists, and the PCO specifies a username@domain (see Figure 13-1). This format is used to implement virtual APN support through a virtual APN. In this scenario the APN IE designates a virtual access point. The APN IE is used to direct the request to GDM (the SGSN's DNS query for the *virtual* APN should return the IP address of GDM). GDM uses the domain as the real APN, and performs a DNS query on the domain name to locate the appropriate destination GGSN.

- **CreatePDPContext (APN=real, PCO=ciscouser)**

  In this format the APN IE exists, and the PCO only specifies a username (no domain). In this scenario the APN IE must designate a real access point. The APN IE is used to direct the request to GDM (the SGSN's DNS query for the *real* APN should return the IP address of GDM). GDM also uses the APN IE (the real APN) to perform a DNS query to locate the appropriate destination GGSN.

- **CreatePDPContext (APN=real, PCO=)**

  In this format, the APN IE exists, and the PCO is null. This format is found when anonymous access is being used. In this scenario the APN IE must designate a real access point. The APN IE is used to direct the request to GDM (the SGSN's DNS query for the *real* APN should return the IP address of GDM). GDM also uses the APN IE (the real APN) to perform a DNS query to locate the appropriate destination GGSN.

Load Balancing Processing by GDM

GDM supports basic load balancing using the DNS server's ability to return to GDM a list of IP addresses in round-robin fashion for a particular domain name. The DNS server can return a list of up to 8 addresses to GDM for each domain name. GDM always uses the first IP address returned by the DNS server. The IP addresses correspond to the GGSNs available to support the requested real APN.

The name for which GDM performs a DNS query is based upon the content of the APN IE and the PCO IE of the create PDP context request. If the create PDP context request specifies a domain, then GDM queries the DNS server for that domain name. The username and password that the MS requests in the form of "login@domain" is provided in the PCO IE of the create PDP context request. The domain name in the MS request, and for which GDM queries the DNS server, should correspond to the name of the real APNs that are configured on the GGSNs. Those GGSNs provide connectivity to the physical network for that APN.

If the MS does not specify a domain, or the PCO is null, then GDM queries the DNS server for the real APN found in the APN IE.

For load balancing support, you must configure the DNS server for GDM with the IP addresses of all of the GGSNs (up to 8) that support connectivity to the physical network for the domain. You also should verify that the round-robin mechanism is enabled for the DNS server when it returns a list of IP addresses for a domain.

Benefits

GDM provides the following benefits:

- Reduction in HLR provisioning through virtual APN support.
- Sharing of GGSN resources using round-robin load balancing.
- Backup router support for GDM functions using HSRP.

Planning to Configure GDM

This chapter describes information that you should know as you prepare to configure the GDM in the Cisco IOS GGSN software. It provides information about important planning and configuration dependencies for the other entities in the GPRS/UMTS network that support the GDM environment.

This chapter includes the following sections:

Prerequisites

Successful implementation of GDM involves proper configuration of the Cisco Systems router as well as proper configuration of some other network entities within the GPRS/UMTS PLMN to support the GDM environment.

This section describes the following other planning aspects and configuration in the GPRS/UMTS network that you need to consider before configuring GDM:

Planning Access Points

GDM supports the use of both real and virtual access points to access destination networks through one or more GGSNs.

Real access points always direct access to a physical, target network. A virtual access point name (APN) is a non-physical entity used by Cisco Systems to represent an access point that does not itself provide direct access to a real target network. Cisco Systems provides virtual APN support in the GPRS/UMTS PLMN.

You can use virtual access points to consolidate access to multiple, physical target networks through a single access point. GDM always uses real access points to direct PDP contexts to an external network through a GGSN. Therefore, virtual access points should be used in combination with real access points on a GGSN.

To implement virtual APN support using GDM, you need to determine the name(s) of the virtual access point(s) that you want subscribers to use for access to one or more real APNs that are configured on your GGSNs.

GDM determines how to process a PDP context according to the content found in the APN Information Element (IE) and the Protocol Configuration Option (PCO) of the create PDP context request. For more information about GDM's request processing and how GDM processes different request scenarios, see the "Request Processing by GDM" section on page 13-2 in the "Overview of GDM" chapter.

What's Next

Once you determine your real and virtual APNs, you need to be sure that the APN information is properly implemented in other areas of the network by performing the following tasks:

- Provision the HLR.
- Configure the DNS server that provides support to the SGSN and the DNS server that provides support to GDM.
- If you are implementing virtual APN support, inform subscribers that they need to specify the appropriate virtual APN when requesting access to the network. By specifying the virtual APN, and by fully qualifying their username in their initial request in the form of login@domain, these subscribers are allowed access to the real target networks through the GGSNs accessible by GDM.

Provisioning the HLR

If you plan to support both real and virtual access points for create PDP context requests, then you will need to provision the HLR with both the real and virtual APNs for the applicable subscribers.

However, if you are using virtual APN support, you do not need to specify the names of each of the real APNs in the HLR's subscriber information, as long as those real APNs are available through GDM. Therefore, when support of additional network access points are added to GGSNs in the GDM environment using a virtual APN, there is no need to do any further provisioning of the HLR subscriber information. This is the primary benefit for using virtual APN support.

After you determine the name of the virtual access point that you want to support, you need to provision the HLR with the virtual APN information for the applicable subscribers. You need to specify the virtual APN for those subscribers with permission to reach any of the real target networks that are accessible from the GGSNs supported by GDM using virtual APN.

To reach different networks, users continue to specify the same virtual APN, but designate the real target network by fully qualifying their username in the form of login@domain, where domain is the name of the real APN, or target network.

Configuring DNS Servers

To support GDM, there are two aspects of DNS server support that you need to configure:

- Configuring the DNS Server for the SGSN, page 14-3
- Configuring the DNS Server for GDM, page 14-3

Configuring the DNS Server for the SGSN

When the SGSN receives a real or virtual APN in the APN IE of a create PDP context request and verifies the user's subscription information with the HLR for that APN, the SGSN queries a DNS server for the IP address associated with that APN. For GDM to work properly, the DNS server must resolve the real and virtual APNs to the IP address of the virtual template interface that you have configured on GDM.

Therefore, at the SGSN DNS server, you must define the IP address of the virtual template interface on GDM for each of the real and virtual APNs that you want to support for MS requests.

For more information about configuring the virtual template interface, see the "Configuring the Virtual Template Interface on GDM" section on page 15-2.

Configuring the DNS Server for GDM

GDM uses a DNS server to obtain the IP addresses of the GGSNs that support connectivity to the target network specified in the create PDP context request. Therefore, you must configure the GDM DNS server (such as the Cisco Network Registrar) to return the IP address of the virtual template interface for one or more GGSNs that provide connectivity to the real APNs that you want to support.

When GDM receives a create PDP context request using virtual APN support, it looks at the domain specified in the username of the protocol configuration option (PCO) information element (IE) to determine the real target network. Then, GDM queries the DNS server to provide the IP addresses of the GGSNs for that domain. In this case, GDM replaces the original content of the APN field so that it now contains the APN of the target network (as specified in the PCO IE), not the virtual APN. Once GDM receives an IP address from the DNS server, it sends the create PDP context request to the GGSN at the IP address returned by DNS.

For other create PDP context requests where no domain is specified, or the PCO IE is null, GDM uses the value of the APN IE as the real APN for its DNS query.

Configuring a Route From the SGSN to GDM

To reach GDM, you must be sure that the SGSN can successfully route to the IP address of the virtual template interface on GDM.

If you are configuring multiple GDMs using Cisco Systems Hot Standby Router Protocol (HSRP), you should configure each GDM with the same IP address at the virtual template interface. Then, you should be sure that the SGSN can route the IP address of the virtual template interface through the LAN segment where you are running HSRP, as specified by the **standby ip** interface configuration command.

For more information about configuring GDM, see the Chapter 15, "Configuring GDM."

Implementing Multiple GDM Routers

You can implement multiple GDM routers to provide backup services in the GPRS/UMTS PLMN. To do this, you can use the HSRP.

There are certain configuration restrictions that you need to follow to support HSRP with GDM:

- Each GDM router must be configured with the same IP address for the virtual template interface.
- Each GDM router must use the same standby IP address, and be in the same HSRP group. GDM does not support multiple HSRP groups.
- The SGSN must be able to route the IP address of the GDM virtual template interface to the IP address that you specified in the **standby ip** interface configuration command.

For information about the configuration commands to use to configure HSRP on GDM, see the "Configuring HSRP on GDM" section on page 15-6. For a configuration example using HSRP, see the "GDM Configuration Example" section on page 15-11.

For more detailed information about HSRP in the Cisco IOS software, refer to the *Cisco IOS IP Configuration Guide*.

Restrictions

When implementing GDM, be aware of the following configuration restrictions:

- GDM supports a single HSRP group only. Therefore, a GDM router cannot be configured to support more than one HSRP group.
- GDM cannot simultaneously support GGSN functions. Although the GDM functionality is part of the Cisco IOS GGSN software, a Cisco Systems router cannot perform the functions of both a GGSN and GDM at the same time.

Supported Platforms

- Cisco 7206 router
- Cisco 7206 VXR NPE-300 router
- Cisco 7206 VXR NPE-400 router

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by GDM.

MIBs

- CISCO-GTP-DIRECTOR-MIB

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

No new or modified RFCs are supported by GDM.

Related Documents

- *Cisco IOS Interface Configuration Guide*, Release 12.2
- *Cisco IOS Interface Command Reference*, Release 12.2
- *Cisco IOS IP Configuration Guide*, Release 12.2
- *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services*, Release 12.2
- *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*, Release 12.2
- *Cisco IOS IP Command Reference, Volume 3 of 3: Multicast*, Release 12.2
- *Cisco IOS Switching Services Configuration Guide*, Release 12.2
- *Cisco IOS Switching Services Command Reference*, Release 12.2

Configuring GDM

This chapter describes how to configure a Cisco Systems router to provide GDM services. It provides information about configuring access from GDM to the SGSN and GGSN in the GPRS/UMTS network, and configuring DNS services to support the GDM environment. It also provides basic configuration information about establishing a single HSRP group in an environment where multiple GDMs are implemented.

For a complete description of the GGSN GDM commands in this chapter, refer to the *Cisco IOS Mobile Wireless Command Reference*. To locate documentation of other commands that appear in this chapter, you can use the command reference master index or search online. See the "Related Documents" section on page 14-5 for a list of the other Cisco IOS software documentation that might be helpful while configuring GDM.

This chapter includes the following sections:

GDM Configuration Task List

To successfully configure GDM in the GPRS/UMTS network, you not only need to configure the Cisco Systems router for GDM services, but you also need to configure several entities within the GPRS/UMTS network to support proper communication with GDM. For example, you must establish routing between GDM and the SGSN and GGSN, and you must configure the DNS servers to return the appropriate IP addresses for the GDM environment.

To configure GDM, perform the following tasks:

- Configuring GDM Services, page 15-2 (Required)
- Configuring the Virtual Template Interface on GDM, page 15-2 (Required)
- Configuring the Physical Interfaces on GDM, page 15-3 (Required)
- Configuring Routes on GDM, page 15-4 (Required)
- Configuring HSRP on GDM, page 15-6 (Optional)
- Customizing GDM, page 15-9 (Optional)

Configuring GDM Services

To enable the router to perform GDM functions, you must configure the router to support GTP director services.

Note A router cannot support both GGSN and GDM services simultaneously. Therefore, do not attempt to configure the **service gprs gtp-director** command on the same router that is configured with the **service gprs ggsn** command.

To configure GDM director services, use the following command beginning in global configuration mode:

Command	Purpose
`Router(config)# service gprs gtp-director`	Configures a router for GTP Director Module (GDM) functions.

Configuring the Virtual Template Interface on GDM

The Cisco IOS software uses a logical interface called a virtual template to configure the router for GDM functions. The virtual template interface provides the required IP addressing and GTP encapsulation to support GDM services.

Be sure to meet the following requirements when you configure GDM on a Cisco Systems router:

- Configure only a single virtual template interface with GTP encapsulation on GDM.
- Configure the IP address of the virtual template for GDM on a different network than the physical interfaces that are configured on the router.

Note If you are implementing multiple GDMs in your network using HSRP, configure the same IP address for the virtual template interface on each router.

To configure the GDM virtual template interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`router(config)# interface virtual-template number`	Creates a virtual template interface, where *number* identifies the virtual template interface. This command enters you into interface configuration mode. **Note** The GGSN supports only a single virtual template for the GTP virtual interface.
Step 2	`router(config-if)# ip address ip-address mask [secondary]`	Specifies an IP address for the interface, where: • *ip-address*—Specifies the IP address of the interface in dotted decimal format. **Note** The IP address of the virtual template interface must be on a different network than the physical interfaces on GDM. • *mask*—Specifies a subnet mask in dotted decimal format. • **secondary**—Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 3	`router(config-if)# encapsulation gtp`	Specifies GTP as the encapsulation type for packets transmitted over the virtual template interface.

Configuring the Physical Interfaces on GDM

To properly configure GDM, you must configure the physical interfaces that GDM uses to reach the networks on which the SGSN and GGSNs reside.

The type of physical interface that you configure on GDM depends on whether you are supporting an SGSN or GGSN that is collocated with GDM, or an SGSN or GGSN that is connected remotely through a WAN interface.

For example, when GDM is collocated with the SGSN or GGSN, the physical interface is frequently configured for Fast Ethernet. The supported WAN interfaces for a remote SGSN or GGSN include T1/E1, T3/E3, and Frame Relay.

For more information about configuring physical interfaces on Cisco Systems routers, see the *Cisco IOS Interface Configuration Guide* and the *Cisco IOS Interface Command Reference*.

To configure a physical interface to the SGSN or GGSN that supports Fast Ethernet on a Cisco 7200 series router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type slot/port`	Defines a physical interface on GDM, where *type* is **fastethernet**, and *slot/port* is the hardware slot and port on the interface.
Step 2	`Router(config-if)# ip address ip-address mask [secondary]`	Specifies an IP address for the interface, where: • *ip-address*—Specifies the IP address of the interface in dotted decimal format. • *mask*—Specifies a subnet mask in dotted decimal format. • **secondary**—Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.

Configuring Routes on GDM

To communicate with the SGSN and the GGSNs, you can use static routes or a routing protocol, such as Open Shortest Path First (OSPF). GDM may or may not reside on the same subnet as the SGSN or GGSNs, so you need to adjust your routing configuration according to your particular network configuration and requirements.

Note For the SGSN to communicate successfully with GDM, the SGSN must also configure a static route, or be able to dynamically route to the IP address of the GDM *virtual template,* not the IP address of an GDM physical interface. For more information about the configuration requirements in the other parts of the GPRS/UMTS PLMN, see the Chapter 14, "Planning to Configure GDM."

The following sections provide some basic commands that you can use to configure a static route or enable OSPF routing on GDM. For more information about configuring IP routes and OSPF, see the *Cisco IOS IP Configuration Guide* and *Cisco IOS IP Command References*.

The following topics are included in this section:

Configuring a Static Route on GDM

A static route establishes a fixed route that is stored in the routing table. If you are not implementing a routing protocol, such as OSPF, then you can configure a static route from GDM to the SGSN, or from GDM to the GGSNs, to establish the route that GDM uses to reach these network devices.

When you define a static route from GDM to the SGSN, you need to create the route between a *physical* interface on GDM to the SGSN. However, for the SGSN to properly communicate with GDM, the SGSN must be able to route to the *logical* interface—the virtual template interface—on GDM. For more information about configuring a route from the SGSN to GDM, see the "Configuring a Route From the SGSN to GDM" section on page 14-3.

To configure a static route from a physical interface on GDM to the SGSN or GGSN, use the following command beginning in global configuration mode:

Command	Purpose
`Router(config)# ip route prefix mask {ip-address | interface-type interface-number} [distance] [tag tag] [permanent]`	Configures a static IP route, where: • *prefix*—Specifies the IP route prefix for the destination. • *mask*—Specifies the prefix mask for the destination. • *ip-address*—Specifies the IP address of the next hop that can be used to reach the destination network. • *interface-type interface-number*—Specifies the network interface type and interface number that can be used to reach the destination network. • *distance*—Specifies an administrative distance for the route. • **tag** *tag*—Specifies a tag value that can be used as a "match" value for controlling redistribution via route maps. • **permanent**—Specifies that the route will not be removed, even if the interface shuts down.

Configuring OSPF on GDM

As with other routing protocols, enabling OSPF requires that you create an OSPF routing process, specify the range of IP addresses to be associated with the routing process, and assign area IDs to be associated with that range of IP addresses.

To configure OSPF, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router ospf process-id`	Enables OSPF routing, and enters router configuration mode, where *process-id* specifies an internally used identification parameter for an OSPF routing process. The *process-id* is locally assigned and can be any positive integer. A unique value is assigned for each OSPF routing process.
Step 2	`Router(config-router)# network ip-address wildcard-mask area area-id`	Defines an interface on which OSPF runs and defines the area ID for that interface, where: • *ip-address*—Specifies the IP address to be associated with the OSPF network area. • *wildcard-mask*—Specifies the IP address mask that includes “don't care” bits for the OSPF network area. • *area-id*—Specifies the area that is to be associated with the OSPF address range. It can be specified as either a decimal value or as an IP address. If you intend to associate areas with IP subnets, you can specify a subnet address as the area-id.

Configuring HSRP on GDM

To provide increased network availability, a backup GDM router can be configured to automatically switch over and become the primary GDM router using HSRP. The backup GDM router can provide access to the GGSNs if the primary GDM router, or even a critical interface on the primary GDM router, becomes unavailable.

When configuring GDM for HSRP, be aware of the following restrictions:

- GDM supports a single HSRP group only. Therefore, an GDM router cannot be configured to support more than one HSRP group.
- When you configure the virtual template interface on each GDM router, use the same IP address.
- Configure the same standby IP address on each GDM using the **standby ip** interface configuration command and be sure that the SGSN can route the IP address of the virtual template interface through that LAN segment where you are running HSRP.

You need to configure HSRP on the physical interface of both the primary and backup GDM routers. Typically, these physical interfaces are on the same network, and the **standby ip** address is also on the same network.

For an example of HSRP configuration on GDM, see the “GDM Configuration Example” section on page 15-11.

This section describes some of the basic commands that you can use to implement HSRP on GDM. For more information about HSRP configuration, see the *Cisco IOS IP Configuration Guide*, and *Cisco IOS IP Command Reference, Volume 1 of 3*.

To configure HSRP on GDM, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# standby [group-number] ip [ip-address [secondary]]`	Activates HSRP on the interface, where: • *group-number*—Specifies the HSRP group number for which HSRP is being activated. The default group is 0. • *ip-address*—Specifies the IP address of the Hot Standby router interface. • **secondary**—Indicates the IP address is a secondary Hot Standby router interface. Useful on interfaces with primary and secondary addresses; you can configure primary and secondary HSRP addresses.
Step 2	`Router(config-if)# standby [group-number] priority priority`	Specifies the HSRP priority, which determines which router becomes the active router, where: • *group-number*—Specifies the HSRP group number to which the priority applies. The default group is 0. • *priority*—Specifies a value (between 1 and 255) that prioritizes the router for hot standby. The default value is 100.

	Command	Purpose
Step 3	`Router(config-if)# standby [group-number] preempt [delay [minimum | sync] delay]`	Configures HSRP preemption and preemption delay, which means that when the local router has a Hot Standby priority higher than the current active router, the local router should attempt to assume control as the active router. If the preempt keyword is not configured, the local router assumes control as the active router only if it receives information indicating that there is no router currently in the active state (acting as the designated router). The following options are available: • *group-number*—Specifies the HSRP group number to which the preemption applies. The default group is 0. • **delay minimum** *delay*—Specifies that the local router postpones taking over the active role for delay (minimum) seconds since that router was last restarted. The range is from 0 to 3600 seconds (1 hour). The default is 0 seconds (no delay). • **delay sync** *delay*—Specifies the maximum synchronization period in seconds.
Step 4	`Router(config-if)# standby [group-number] track interface-type interface-number [interface-priority]`	Configures an interface so that the Hot Standby priority changes based on the availability of other interfaces, where: • *group-number*—Specifies the HSRP group number to which the tracking applies. The default group is 0. • *interface-type*—Specifies the interface type (combined with interface number) that will be tracked. • *interface-number*—Specifies the interface number that will be tracked. • *interface-priority*—Specifies the amount by which the Hot Standby priority for the router is decremented (or incremented) when the interface goes down (or comes back up). The default value is 10.

Customizing GDM

This section describes options that you can modify to customize your GDM configuration. It includes the following topic:

- Configuring the Retry Timeout Period on GDM, page 15-9

Configuring the Retry Timeout Period on GDM

You can specify the length of time during which GDM forwards all retries of create PDP context requests for a specific TID from an SGSN to the same GGSN. The retry-timeout value represents the maximum period of time during which GDM expects the real GGSN to establish or reject the PDP context request. The default value is 30 seconds.

It is recommended that the retry-timeout value be specified according to the following formula:

$T \geq (N3 \bullet T3 + B)$,

where:

- T is the GDM retry-timeout. This is the value that you need to determine for the **gprs gtp-director retry-timeout** command on the GDM router.
- N3 is the retry count that is configured on the SGSN.
- T3 is the retry timer that is configured on the SGSN.
- B is some integer that you choose as a buffer factor. The buffer factor is suggested to allow sufficient time for routing and processing the request by the real GGSN.

Note You can configure the **gprs gtp-director retry-timeout** command in real time for GDM. The new value will be used for create PDP context requests coming in for any new TIDs. The new value is not retroactive for existing TIDs. Therefore, the old value is used for any PDP context requests for an existing TID.

To configure the retry timeout period on GDM, use the following command beginning in global configuration mode:

Command	Purpose
`Router(config)# gprs gtp-director retry-timeout seconds`	Specifies the amount of time during which GDM forwards all retries of create PDP context requests for a specific TID from an SGSN to the same GGSN.

Verifying GDM Configuration

To verify GDM configuration, you can use the **show running-config** privileged EXEC command. The following sample output shows the configuration of a router named "adm1" with HSRP configured. Some of the commands most relevant to the GDM configuration are shown in bold.

Observe that your configuration contains similar output fields as those highlighted in bold in the output below. If you are not using HSRP, then the standby interface configuration commands are not needed:

```
GDM1# show running-config
Building configuration...

Current configuration : 2875 bytes
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service gprs gtp-director
!
hostname adm1
!
ip cef
ip subnet-zero
!
ip name-server 172.16.43.11
!
interface FastEthernet0/0
 description Interface to Backup GDM2 and SGSN
 ip address 10.10.1.3 255.255.255.0
 no ip mroute-cache
 duplex full
 standby ip 10.10.1.10
 standby priority 105
 standby preempt
 standby track FastEthernet2/0
!
interface FastEthernet2/0
 description Interface to GGSN
 ip address 10.10.2.1 255.255.255.0
 no ip mroute-cache
 duplex full
!
interface FastEthernet6/0
 ip address 172.16.43.243 255.255.255.240
 no ip mroute-cache
 duplex half
!
interface Virtual-Template1
 ip address 10.11.11.1 255.255.255.0
 encapsulation gtp
!
router ospf 200
 log-adjacency-changes
 network 10.10.0.0 0.0.255.255 area 0
!
ip default-gateway 172.16.43.241
ip classless
ip route 10.22.22.1 255.255.255.255 FastEthernet2/0
ip route 192.64.0.0 255.0.0.0 172.16.43.241
ip route 172.16.0.0 255.255.0.0 172.16.43.241
no ip http server
```

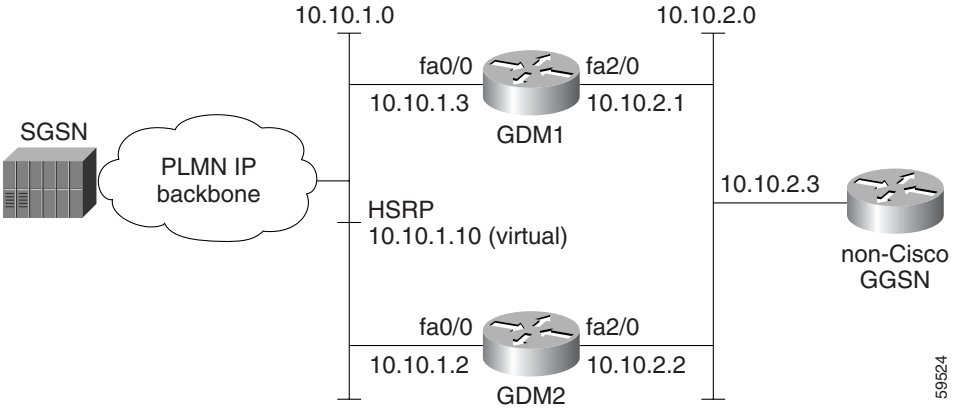
```
no ip pim bidir-enable
!
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
gprs gtp-director retry-timeout 60
!
call rsvp-sync
!
mgcp profile default
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 length 0
line aux 0
line vty 0 4
 password lab
 login
line vty 5 15
 login
!
!
end
```

GDM Configuration Example

The following GDM configuration example shows two routers configured for GDM services using HSRP. Although GDM will likely be used to support multiple GGSNs for load balancing, only a single GGSN is shown in this example for simplicity.

Each GDM router configures the same IP address of 10.11.11.1 for the virtual template interface. In this example, the DNS query by the SGSN should return the IP address of 10.11.11.1 for the GDM router associated with the implemented real or virtual APNs. However, the SGSN must route the address 10.11.11.1 through the LAN segment in use by HSRP (the 10.10.1.0 subnet). The address 10.10.1.10 is the standby IP address that GDM1 and GDM2 routers configure for HSRP support using the **standby ip** interface configuration command.

Because the FastEthernet2/0 interface is the single point of failure to the GGSN in this example, each GDM tracks the status of this interface for HSRP support using the **standby track** interface configuration command.



GDM1 Configuration

```
Current configuration : 2875 bytes
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
! Enable the router for GDM services
!
service gprs gtp-director
!
hostname adm1
!
ip subnet-zero
ip cef
!
ip name-server 172.16.43.11
!
! Configure the physical interface on
! the network to support HSRP and
! also reachable by GDM2 and the SGSN
!
interface FastEthernet0/0
 ip address 10.10.1.3 255.255.255.0
 no ip mroute-cache
 duplex full
!
! Configure HSRP on same network
! reachable by GDM2 and the SGSN
! The standby ip address is the
! same as the address configured on GDM2
!
 standby ip 10.10.1.10
 standby priority 105
 standby preempt
!
! Monitor the availability of
! the FastEthernet2/0 interface
! to the GGSN for standby support
!
 standby track FastEthernet2/0
!
! Configure physical interface
```

```
! to reach the GGSN network
!
interface FastEthernet2/0
 ip address 10.10.2.1 255.255.255.0
 no ip mroute-cache
 duplex full
!
interface FastEthernet6/0
 ip address 172.16.43.243 255.255.255.240
 no ip mroute-cache
 duplex half
!
! Configure the virtual template interface
! to support GTP encapsulation.
! You must configure the same virtual-template
! IP address on both GDM1 and GDM2
! for successful HSRP support
!
interface Virtual-Template1
 ip address 10.11.11.1 255.255.255.0
 encapsulation gtp
!
router ospf 200
 log-adjacency-changes
 network 10.10.0.0 0.0.255.255 area 0
!
ip default-gateway 172.16.43.241
ip classless
ip route 10.22.22.1 255.255.255.255 FastEthernet2/0
ip route 192.64.0.0 255.0.0.0 172.16.43.241
ip route 172.16.0.0 255.255.0.0 172.16.43.241
no ip http server
no ip pim bidir-enable
!
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
! Configure GDM to forward retries of
! create PDP context requests from an
! SGSN for 1 minute
!
gprs gtp-director retry-timeout 60
!
call rsvp-sync
!
mgcp profile default
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 length 0
line aux 0
line vty 0 4
 password lab
 login
line vty 5 15
 login
!
!
end
```

GDM2 Configuration

```
Current configuration : 2452 bytes
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
! Enable the router for GDM services
!
service gprs gtp-director
!
hostname adm2
!
ip cef
ip subnet-zero
!
ip name-server 172.16.43.11
!
! Configure the physical interface on
! the network to support HSRP and
! also reachable by GDM1 and the SGSN
!
interface FastEthernet0/0
 ip address 10.10.1.2 255.255.255.0
 no ip mroute-cache
 duplex half
!
! Configure HSRP on same network
! reachable by GDM1 and the SGSN
! The standby ip address is the
! same as the address configured on GDM1
!
 standby ip 10.10.1.10
 standby priority 100
 standby preempt
!
! Monitor the availability of
! the FastEthernet2/0 interface
! to the GGSN for standby support
!
 standby track FastEthernet2/0
!
! Configure physical interface
! to reach the GGSN network
!
interface FastEthernet2/0
 ip address 10.10.2.2 255.255.255.0
 no ip mroute-cache
 duplex half
!
interface FastEthernet6/0
 ip address 172.16.43.249 255.255.255.240
 no ip mroute-cache
 duplex half
!
! Configure the virtual template interface
! to support GTP encapsulation
! You must configure the same virtual-template
! IP address on both GDM1 and GDM2
! for successful HSRP support
!
```

```
interface Virtual-Template1
 ip address 10.11.11.1 255.255.255.0
 encapsulation gtp
!
router ospf 300
 log-adjacency-changes
 network 10.10.0.0 0.0.255.255 area 0
!
ip default-gateway 172.16.43.241
ip classless
ip route 10.22.22.1 255.255.255.255 FastEthernet2/0
ip route 192.64.0.0 255.255.0.0 172.16.43.241
no ip http server
no ip pim bidir-enable
!
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
call rsvp-sync
!
mgcp profile default
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 login
!
!
end
```

Monitoring and Maintaining GDM

This chapter describes the commands used to display GDM statistics and pending requests. For a complete description of the GGSN GDM commands in this chapter, refer to the *Cisco IOS Mobile Wireless Command Reference*.

This chapter includes the following sections:

Show Command Summary

This section provides a list of the **show** commands available for GDM. The following privileged EXEC commands are used to monitor and maintain GDM:

Command	Purpose
`Router# show gprs gtp-director pending-request`	Displays a list of the create PDP context requests sent by GDM to a real GGSN that are pending expiration of the retry timer.
`Router# show gprs gtp-director statistics`	Displays the current statistics for create requests received by GDM.

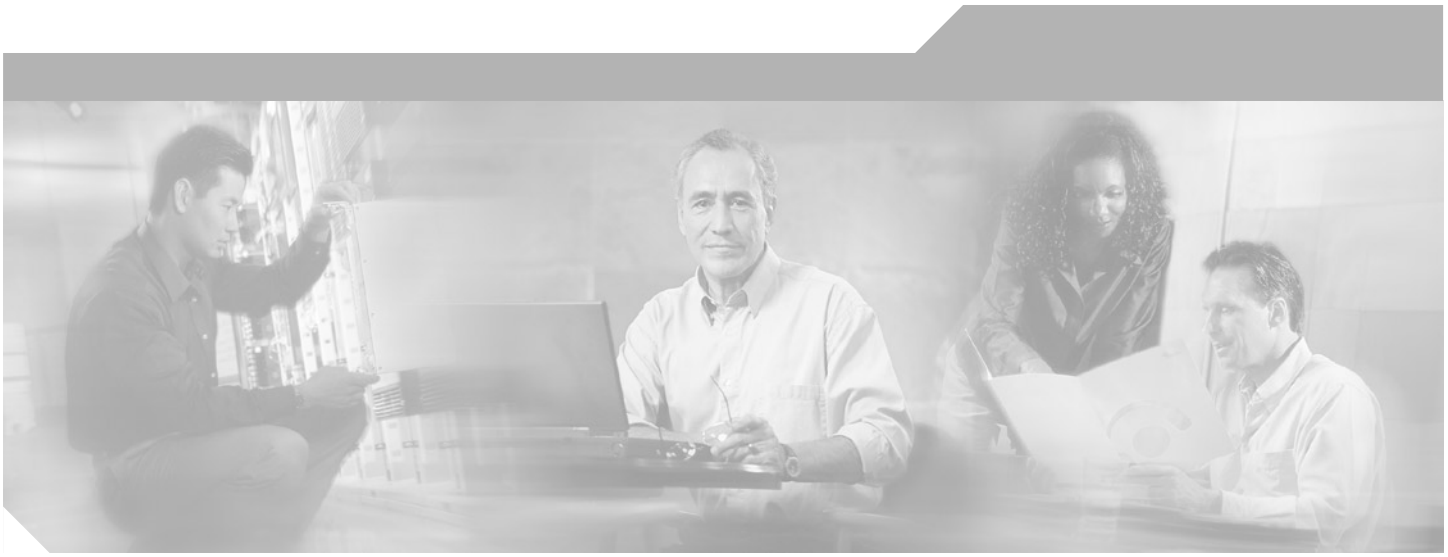
Displaying Pending Requests

When you are monitoring GDM, it is important to recognize that GDM does not itself track the success or failure of create PDP context requests that it forwards to the GGSNs on behalf of an SGSN. GDM only participates in the forwarding of create PDP context requests to a GGSN for a specified length of time (default is 30 seconds) for a particular TID.

Therefore, when you use the **show gprs gtp-director pending-request** command, you are seeing only those PDP context requests of TIDs for which GDM is still accepting the forwarding of retries to the GGSN. Once the retry period has expired, the TID will no longer appear in the pending request display. However, the PDP context request may or may not have been activated between the SGSN and GGSN.

To verify the success of the PDP context request, you must monitor the path between the SGSN and GGSN, or monitor the GGSN itself. You can use the **show gprs gtp-director pending-request** command to display the IP addresses of the SGSN and GGSN for a particular TID.

CISCO SYSTEMS



PART 3

Appendix

Glossary

1G mobile network—first generation mobile network. Refers to the initial category of mobile wireless networks that use analog technology only. Advanced Mobile Phone Service (AMPS) is an example of a 1G mobile network standard.

2G mobile network—second generation mobile network. Refers generically to a category of mobile wireless networks and services that implement digital technology. GSM is an example of a 2G mobile network standard.

2G+ mobile network—second generation plus mobile network. Refers generically to a category of mobile wireless networks that support higher data rates than 2G mobile networks. GPRS is an example of a 2G+ mobile network standard. Also called 2.5G.

3G mobile network—third generation mobile network. Refers generically to a category of next-generation mobile networks such as UMTS and IMT-2000.

APN—access point name. Logical name composed of a network ID (mandatory) and operator ID (optional) that identifies a PDN or private network that is configured on and accessible from a GGSN in a GPRS network. An APN corresponds to the DNS name of a GGSN.

BSC—base station controller. Provides the control functions and physical links between the MSC and BTS in a GSM mobile wireless network. The BSC controls the interface between the SGSN and the BTS in a GPRS network. The BSC is a high-capacity telephony switch that provides handoff functions, cell configuration data, and controls radio frequency power levels in BTSs. The combined functions of the BSC and BTS are referred to as the BSS.

BSS—base station subsystem. Refers to the radio-related functions provided by the BTS and BSC in a GSM mobile wireless network.

BTS—base transceiver station. A land-based station in a GSM mobile wireless network that consists of transceivers and antennas, which handle the radio interface to a mobile station. One or more BTSs are controlled by a BSC. The combined functions of the BTS and BSC are referred to as the BSS.

CDMA—code division multiple access. A method of dividing a radio spectrum to be shared by multiple users through the assignment of unique codes. CDMA implements spread spectrum transmission.

CDR—call detail record. Used in the original telephony networks, and now extended to mobile wireless network calls, the CDR contains billing information for charging purposes. In a GPRS network, the charging gateway sends the billing information within a CDR to the network service provider for that subscriber.

GGSN—gateway GPRS support node. A GPRS network entity that serves as the mobile wireless gateway between an SGSN and PDNs. The GGSN allows mobile users to access PDNs.

Gi interface—Reference point between a GPRS network and an external packet data network.

Gn interface—Interface between GSNs within the same PLMN in a GPRS network. GTP is a protocol defined on both the Gn and Gp interfaces between GSNs in a GPRS network.

Gp interface—Interface between GSNs within different PLMNs in a GPRS network. GTP is a protocol defined on both the Gp and Gn interfaces between GSNs in a GPRS network.

GPRS—General Packet Radio Service. An ETSI standard that defines the implementation of packet data services on a GSM network.

GSM—Global System for Mobile Communication. A second generation (2G) mobile wireless networking standard defined by ETSI, GSM is widely deployed throughout the world. GSM uses TDMA technology and operates in the 900-MHz radio band.

GSN—GPRS support node. GSN (or GSNs) refers to the general functions of a group of both GGSNs and SGSNs in a GPRS network.

GTP—GPRS tunneling protocol. GTP handles the flow of user packet data and signaling information between the SGSN and GGSN in a GPRS network. GTP is defined on both the Gn and Gp interfaces of a GPRS network.

GTP tunnel—Used to communicate between an external packet data network and a mobile station in a GPRS network. A GTP tunnel is referenced by an identifier called a TID and is defined by two associated PDP contexts residing in different GSNs. A tunnel is created whenever an SGSN sends a Create PDP Context Request in a GPRS network.

HLR—home location register. A database that contains information about subscribers to a mobile network. The HLR registers subscribers for a particular service provider. The HLR stores “permanent” subscriber information (rather than temporary subscriber data, which a VLR manages), including the service profile, location information, and activity status of the mobile user.

IMSI—international mobile subscriber identity. A unique identifier stored in the SIM of a mobile station. The MS sends the IMSI to a BTS for identification of the MS in the GSM network. The BTS looks for the IMSI in the HLR.

MCC—mobile country code. Part of the IMSI that uniquely identifies the home country of the mobile station.

MNC—mobile network code. Two or three-digit number within the NMSI part of the IMSI that uniquely identifies the home PLMN of the mobile station.

MS—mobile station. Refers generically to any mobile device, such as a mobile handset or computer, that is used to access network services. GPRS networks support three classes of MS, which describe the type of operation supported within the GPRS and GSM mobile wireless networks. For example, a Class A MS supports simultaneous operation of GPRS and GSM services.

MSC—mobile switching center. Provides telephony switching services and controls calls between telephone and data systems.

MSIN—mobile station identification number. Part of the mobile station identification following the MNC that uniquely identifies the mobile station within a PLMN.

MSISDN—MS international PSTN/ISDN.

NMSI—national mobile station identity. Part of the IMSI that uniquely identifies the mobile station nationally. The NMSI consists of the MNC followed by the MSIN.

PCU—packet control unit. A network component that normally resides in a BSC and directs packet traffic to the SGSN for processing by the GPRS network.

PDN—public/private/packet data network. Represents a public or private packet-based network, such as an IP or X.25 network.

When used to represent a public data network, PDN refers to a network operated either by a government (as in Europe) or by a private concern to provide computer communications to the public, usually for a fee. PDNs enable small organizations to create a WAN without all the equipment costs of long-distance circuits.

PDP—packet data protocol. Network protocol used by external packet data networks that communicate with a GPRS network. IP is an example of a PDP supported by GPRS.

PDP context—Refers to a set of information (such as a charging ID) that describes a mobile wireless service call or session, which is used by mobile stations and GSNs in a GPRS network to identify the session.

PDU—protocol data unit. OSI term for packet. See also Bpdu and packet .

PLMN—public land mobile network. Generic name for all mobile wireless networks that use earth-based stations rather than satellites. PLMN is the mobile equivalent of the PSTN.

SGSN—serving GPRS support node. A GPRS network entity that sends data to and receives data from mobile stations, and maintains information about the location of an MS. The SGSN communicates between the MS and the GGSN; the GGSN provides access to the data network.

SIM—subscriber identity module. Component of an MS in a GSM network that contains all of the subscriber information.

SMG—Special Mobile Group. A standards body within ETSI that develops specifications related to mobile networking technologies such as GSM and GPRS.

TDMA—time division multiple access. A method of dividing a transmission channel to be shared by multiple users through the assignment of time slots. See also TDM.

TID—tunnel identifier. Used to identify a GTP tunnel between two GSNs in a GPRS network. Contains an MM Context ID and an NSAPI. A tunnel is created whenever an SGSN sends a Create PDP Context Request in a GPRS network. See also GTP tunnel.

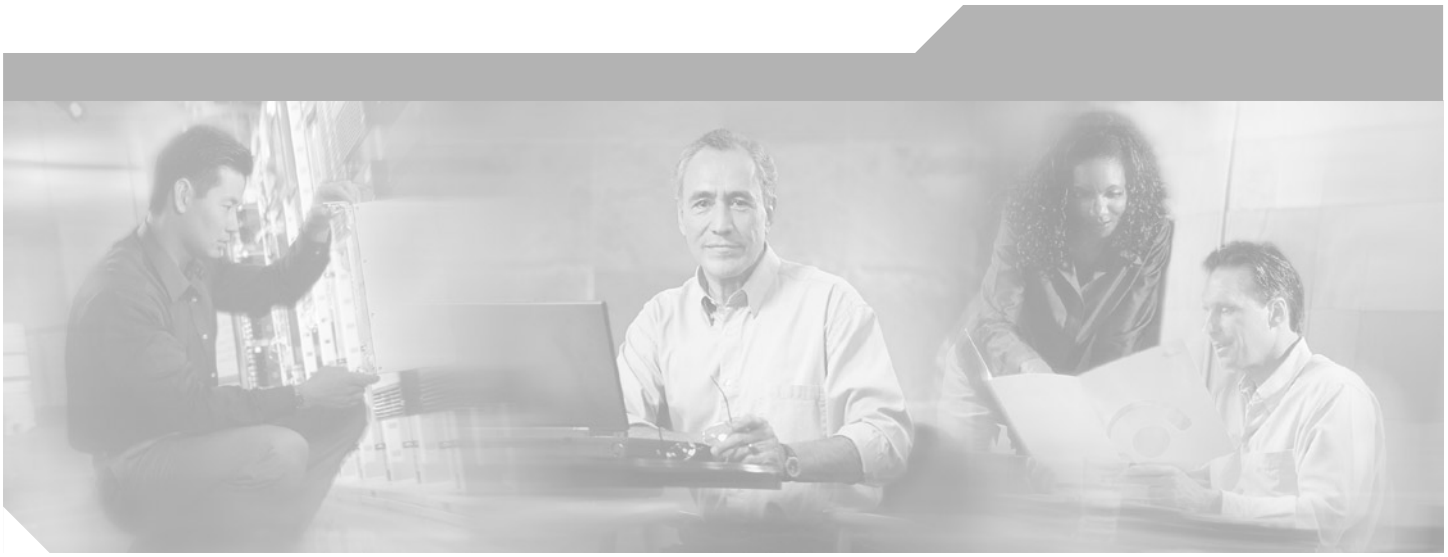
TMSI—temporary mobile subscriber identity. A temporary code used to identify an MS, which is assigned using encryption after the MS is identified to the HLR.

UMTS—Universal Mobile Telephone Service. A 3G mobile wireless telecommunications system whose standards are being developed by the Third Generation Partnership Project (3GPP).

VLR—visitor location register. A database that contains temporary information about subscribers who roam into an area controlled by another MSC. The VLR communicates with the HLR of the subscriber to request data about that subscriber.

VRF—A VPN routing/forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

CISCO SYSTEMS



PART 4

Index

Symbols

A

B

C

D

E

F

G

H

I

L

M

N

P

Q

R

S

T

U

V