CHAPTER 7

Configuring Network Access to the GGSN

This chapter describes how to configure access from the gateway GPRS support node (GGSN) to a serving GPRS support node (SGSN), public 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 GGSN Command Reference* for the Cisco GGSN release you are using. 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 an Interface to the SGSN

To establish access to an SGSN, you must configure an interface to the SGSN. In general packet radio service/Universal Mobile Telecommunication System (GPRS/UMTS), the interface between the GGSN and the SGSN is referred to as the *Gn interface*. GGSN Release 4.0 and later supports both a 2.5G and 3G Gn interface.

On the Cisco 7600 series router platform, this interface is logical one (on which IEEE 802.1Q encapsulation has been configured) to the Layer 3 routed Gn VLAN configured on the supervisor engine.

For more information about the Gn VLAN on the supervisor engine, see Platform Prerequisites, page 1-2.

For more information about configuring interfaces, see the *Cisco IOS Interface Configuration Guide* and the *Cisco IOS Interface Command Reference*.

Configuring 802.1Q-Encapsulated Subinterfaces

To configure a subinterface that supports IEEE 802.1Q encapsulation to the Gn VLAN, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface gigabitethernet slot/port.subinterface-number`	Specifies the subinterface on which IEEE 802.1Q will be used.
Step 2	`Router(config-if)# encapsulation dot1q vlanid`	Defines the encapsulation format as IEEE 802.1Q (dot1q), and specifies the VLAN identifier.
Step 3	`Router(config-if)# ip address ip-address mask`	Sets a primary IP address for an interface.

Verifying the Interface Configuration to the SGSN

Step 1 To verify that you have properly configured a Gn interface on the supervisor engine, use the **show running-config** command. The following example is a portion of the output from the command showing the Fast Ethernet 8/22 physical interface configuration (see bold text) as the Gn interface to the SGSN:

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

Current configuration :12672 bytes
!
version 12.x
...
interface FastEthernet8/22
 no ip address
 switchport
 switchport access vlan 302
!
interface Vlan101
 description Vlan to GGSN for GA/GN
 ip address 10.1.1.1 255.255.255.0
!
interface Vlan302
 ip address 40.0.2.1 255.255.255.0
```

Step 2 To verify that the physical interface and the Gn VLAN are available, use the **show interface** command on the supervisor engine. The following example shows that the Fast Ethernet 8/22 physical interface to the charging gateway is up, as is the Gn VLAN, VLAN 101.

```
Sup# show ip interface brief FastEthernet8/22
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet8/22           unassigned      YES unset  up                    up

Sup# show ip interface brief Vlan302
Interface                  IP-Address      OK? Method Status                Protocol
Vlan302                    40.0.2.1        YES TFTP   up                    up

Sup#
```

Step 3 To verify the Gn VLAN configuration and availability, use the **show vlan name** command on the supervisor engine. The following example shows the Gn VLAN Gn_1:

```
Sup# show vlan name Gn_1

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
302  Gn_1                             active    Gi4/1, Gi4/2, Gi4/3, Gi7/1
                                                Gi7/2, Gi7/3, Fa8/22, Fa8/26

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
302  enet  100302     1500  -      -      -        -    -        0      0

Remote SPAN VLAN
----------------
Disabled

Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------
```

Step 4 On the GGSN, to verify that you have properly configured a Gn subinterface to the Gn VLAN, use the **show running-config** command. The following example is a portion of the output from the command showing a Gigabit Ethernet 0/0.2 physical interface configuration as the Gn interface to the charging gateway:

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

Current configuration :7390 bytes
!
! Last configuration change at 16:56:05 UTC Wed Jun 25 2003
! NVRAM config last updated at 23:40:27 UTC Fri Jun 13 2003
!
version 12.3
.....
interface GigabitEthernet0/0.2
 description Ga/Gn Interface
 encapsulation dot1Q 101
 ip address 10.1.1.72 255.255.255.0
 no cdp enable
!
.....
ip route 40.1.2.1 255.255.255.255 10.1.1.1
```

Step 5 To verify that the subinterface is available, use the **show ip interface brief** command. The following example shows that the Gigabit Ethernet 0/0.2 subinterface to the Gn VLAN is in “up” status and that the protocol is also “up”:

```
GGSN# show ip interface brief GigabitEthernet0/0.2
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0.2       10.1.1.72       YES NVRAM  up                    up
```

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 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 to 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 to the SGSN, to establish the path between network devices.

To configure a static route from an interface 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 an 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

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.

Note On the Cisco 7600 series router platform, the OSPF routing process is configured on the supervisor engine to advertise only the GPRS tunneling protocol (GTP) server load balancing (SLB) virtual server and the GGSN virtual template 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 supervisor engine configuration, use the **show running-config** command and verify the route that you configured to the SGSN. The following example shows a partial configuration of a configuration to the SGSN:

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

Current configuration :3642 bytes
!
version 12.3
...
ip slb vserver V0-GGSN
 virtual 10.10.10.10 udp 3386 service gtp
```

```
!
vlan 101
 name Internal_Gn/Ga
!
vlan 302
 name Gn_1
!
vlan 303
 name Ga_1
!
interface FastEthernet8/22
 no ip address
 switchport
 switchport access vlan 302
!
interface FastEthernet8/23
 no ip address
 switchport
 switchport access vlan 302
!
interface FastEthernet8/24
 no ip address
 switchport
 switchport access vlan 303
!
interface Vlan101
 description Vlan to GGSN for GA/GN
 ip address 10.1.1.1 255.255.255.0
!
interface Vlan302
 ip address 40.0.2.1 255.255.255.0
!
interface Vlan303
 ip address 40.0.3.1 255.255.255.0
!
router ospf 300
 log-adjacency-changes
 summary-address 9.9.9.0 255.255.255.0
 redistribute static subnets route-map GGSN-routes
 network 40.0.2.0 0.0.0.255 area 300
 network 40.0.3.0 0.0.0.255 area 300
!
ip route 9.9.9.42 255.255.255.255 10.1.1.42
ip route 9.9.9.43 255.255.255.255 10.1.1.43
ip route 9.9.9.44 255.255.255.255 10.1.1.44
ip route 9.9.9.45 255.255.255.255 10.1.1.45
ip route 9.9.9.46 255.255.255.255 10.1.1.46
ip route 9.9.9.72 255.255.255.255 10.1.1.72
ip route 9.9.9.73 255.255.255.255 10.1.1.73
ip route 9.9.9.74 255.255.255.255 10.1.1.74
ip route 9.9.9.75 255.255.255.255 10.1.1.75
ip route 9.9.9.76 255.255.255.255 10.1.1.76
!
access-list 1 permit 9.9.9.0 0.0.0.255
!
route-map GGSN-routes permit 10
 match ip address 1
```

Step 2 To verify the GGSN configuration, use the **show running-config** command. The following example shows a partial configuration of a configuration to the SGSN:

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

Current configuration :3642 bytes
!
version 12.3
!
...

interface GigabitEthernet0/0
 no ip address
!

interface GigabitEthernet0/0.2
 description Ga/Gn Interface
 encapsulation dot1Q 101
 ip address 10.1.1.72 255.255.255.0
 no cdp enable
!
ip route 40.1.2.1 255.255.255.255 10.1.1.1
ip route 40.2.2.1 255.255.255.255 10.1.1.1
ip route 40.1.3.10 255.255.255.255 10.1.1.1
ip route 40.2.3.10 255.255.255.255 10.1.1.1
```

Step 3 To verify that the supervisor engine has established a route to the SGSN, use the **show ip route** command as shown in bold in the following examples:

```
Sup# show ip route ospf 300
9.0.0.0/8 is variably subnetted, 12 subnets, 2 masks
O       9.9.9.0/24 is a summary, 1w1d, Null0
!

Sup# show ip route 9.9.9.72
Routing entry for 9.9.9.72/32
  Known via "static", distance 1, metric 0
  Redistributing via ospf 300
  Routing Descriptor Blocks:
  * 10.1.1.72
      Route metric is 0, traffic share count is 1
!
```

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 on an access point are discussed in the "Configuring Dynamic Addressing 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. In the GPRS/UMTS backbone, the APN 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 public land mobile network (PLMN) in which a GGSN is located. The operator ID contains three decimal-separated labels; 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 it corresponds to the Domain Name System (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 GGSN

Configuring access points is one of the central configuration tasks on the Cisco 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 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 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 GGSN configuration commands use the index number to reference an APN.
- Access group—An additional level of router 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 support the following access point types:

- Real—Uses real access point types to configure the GGSN for direct access to a particular target network through an interface. The GGSN always uses real access points to reach an external network.

  For information on configuring real access points on the GGSN, see the "Configuring Real Access Points on the GGSN" section on page 7-11.

- Virtual—Uses virtual access point types to consolidate access to multiple target networks through a virtual APN access point at the GGSN. Because the GGSN always uses real access points to reach an external network, virtual access points should be used in combination with real access points on the GGSN.

  For information on configuring virtual access points on the GGSN, see the "Configuring Virtual Access Points on the GGSN" section on page 7-32.

Note GGSN Release 1.4 and earlier only support real access points. To address provisioning issues in the PLMN, GGSN Release 3.0 and later support virtual access point types. Additionally, with GGSN Release 6.0, Cisco IOS Release 12.3(14)YU and later, you can configure virtual APNs to be dynamically mapped, per user, to the target APN during a "pre-authentication" phase. For more information, see the "Configuring Virtual Access Points on the GGSN" section on page 7-32.

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 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 7-10 (Required)
- Creating an Access Point and Specifying Its Type on the GGSN, page 7-10 (Required)

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 command, 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 on an access point are summarized in the "Configuring Additional Real Access Point Options" section on page 7-20.

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 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, home location register (HLR), and DNS server.
Step 4	`Router (config-access-point)# access-type {virtual [pre-authenticate [default-apn apn-name]]| 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. Optionally, can be configured to be dynamically mapped, per user, to a target APN. • **real**—APN type that corresponds to an interface to an external network on the GGSN. This is the default value. **Note** The default access-type is real. Therefore, you only need to configure this command if the APN needs to be a virtual access point.

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 an 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 7-12
- VPN Access Using VRF Configuration Task Lists, page 7-13

PDN Access Configuration Task List

Configuring a connection to a public PDN 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 establish access to a PDN in the GPRS/UMTS network, you must configure an interface on the GGSN to connect to the PDN. This interface is referred to as the *Gi interface*.

On the Cisco 7600 series router platform, this interface is a logical one (on which IEEE 802.1Q encapsulation has been configured) to a Layer 3 routed Gi VLAN configured on the supervisor engine.

For more information about the Gi VLAN on the supervisor engine, see "Platform Prerequisites" section on page 1-2.

For more information about configuring interfaces, see the *Cisco IOS Interface Configuration Guide* and the *Cisco IOS Interface Command Reference*.

Note If you are using VPN routing and forwarding (VRF) for VPN access, you must enable Cisco Express Forwarding (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 802.1Q-Encapsulated Subinterfaces

To configure a subinterface that supports IEEE 802.1Q encapsulation to the Gi VLAN, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface gigabitethernet slot/port.subinterface-number`	Specifies the subinterface on which IEEE 802.1Q will be used.
Step 2	`Router(config-if)# encapsulation dot1q vlanid`	Defines the encapsulation format as IEEE 802.1Q (dot1q), and specifies the VLAN identifier.
Step 3	`Router(config-if)# ip address ip-address mask`	Sets a primary IP address for an 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.
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 an interface to an external network on the GGSN. Real is the default value.

For an example of a GPRS access point configuration, see the "Access Point List Configuration Example" section on page 7-51.

VPN Access Using VRF Configuration Task Lists

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

Note VRF is not supported for IPv6 PDPs. Therefore, if the **ipv6** command is configured on an APN on which VRF is enabled, the IPv4 PDPs are routed in VRF, but the IPv6 PDPs are routed in the global routing table.

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

Note VRF is not supported on the Cisco 7600 Supervisor II / MSFC2; therefore, if using the Supervisor II, you must tunnel encapsulated VRF traffic through the Supervisor via a generic routing encapsulation (GRE) tunnel from the GGSN to the PDN. For more information on configuring a tunnel, see the "Configuring Access to a VPN With a Tunnel" section on page 7-18.

The Cisco 7600 Sup720 supports VRF.

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

- Enabling CEF Switching, page 7-14 (Required)
- Configuring a VRF Routing Table on the GGSN, page 7-14 (Required)
- Configuring a Route to the VPN Using VRF, page 7-14 (Required)

- Configuring an Interface to a PDN Using VRF, page 7-16 (Required)
- Configuring Access to a VPN, page 7-17 (Required)

For sample configurations, see the "VRF Tunnel Configuration Example" section on page 7-51.

Enabling CEF Switching

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 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 commands, 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. • **permanent**—Specifies that the route will not be removed, even if the interface shuts down. • **tag** *tag*—Specifies a tag value that can be used as a "match" value for controlling redistribution via route maps.

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:

```
GGSN# 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 establish access to a PDN, an interface on the GGSN to connect to the PDN. This interface is referred to as the Gi interface.

On the Cisco 7600 series router platform, this interface is a logical one (on which IEEE 802.1Q encapsulation has been configured) to a Layer 3 routed Gi VLAN configured on the supervisor engine.

For more information about the Gi VLAN on the supervisor engine, see "Platform Prerequisites" section on page 1-2.

For more information about configuring interfaces, see the *Cisco IOS Interface Configuration Guide* and the *Cisco IOS Interface Command Reference*.

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 802.1Q-Encapsulated Subinterfaces

To configure a subinterface that supports IEEE 802.1Q encapsulation to the Gi VLAN, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface gigabitethernet slot/port.subinterface-number`	Specifies the subinterface on which IEEE 802.1Q will be used.
Step 2	`Router(config-if)# encapsulation dot1q vlanid`	Defines the encapsulation format as IEEE 802.1Q (dot1q), and specifies the VLAN identifier.
Step 3	`Router(config-if)# ip address ip-address mask`	Sets a primary IP address for an interface.

Configuring Access to a VPN

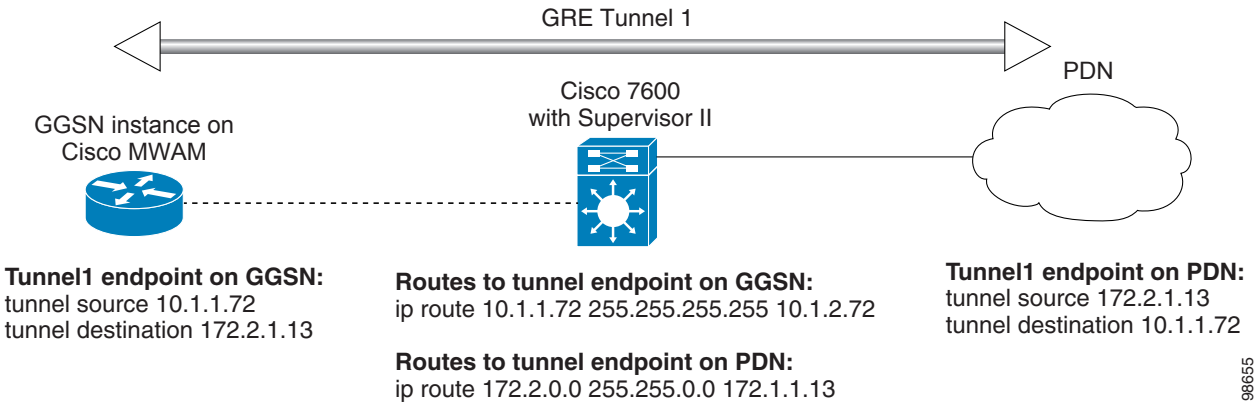
After you have completed the prerequisite configuration tasks, you can configure access to a VPN with a tunnel or without a tunnel.

VRF is not supported on the Cisco 7600 Supervisor II / MSFC2; therefore, if using the Supervisor II, you must tunnel encapsulated VRF traffic through the Supervisor via a generic routing encapsulation (GRE) tunnel from the GGSN to the PDN.

Note The Cisco 7600 Sup720 supports VRF.

Figure 7-1 is a logical view of a GRE tunnel configured between the VRF-aware GGSN and PDN, which tunnels the encapsulated VRF information through the "VRF-unaware" supervisor engine.

Figure 7-1 Tunnel Configuration from the GGSN to PDN through the Cisco 7600 Supervisor II



The following sections describe the different methods you can use to configure access to a VPN:

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

Note With GGSN Release 5.0 and later, you can assign multiple APNs to the same VRF.

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.

Note The Cisco 7600 Supervisor II / MSFC2 does not support VRF; therefore, you must tunnel VRF traffic through the Supervisor via a GRE tunnel as described in the "Configuring Access to a VPN With a Tunnel" section on page 7-18.

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 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 Domain Name System (DNS) server.
Step 4	`Router(config-access-point)# access-type real`	Specifies an APN type that corresponds to an interface to an external network on the GGSN. Real is the default value.
Step 5	`Router(config-access-point)# vrf vrf-name`	Configures VRF 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 Additional Real Access Point Options" section on page 7-20.

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, or if you are configuring access to a VPN via VRF on the Cisco 7600 series router platform, you can configure an IP tunnel to access those private networks. If using the Supervisor/MSFC2 on the Cisco 7600 series router platform, you configure the tunnel to tunnel the VRF traffic through the supervisor engine.

To configure access to the VPN using a tunnel, 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.
Step 4	`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 5	`Router(config-access-point)# access-type real`	Specifies an APN type that corresponds to an interface to an external network on the GGSN. Real is the default value.
Step 6	`Router(config-access-point)# ip-address-pool {dhcp-proxy-client | radius-client | local pool-name | 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. • **local**—Specifies that a local pool provides the IP address. This option requires configuration of a local pool using the **ip local pool** global configuration command. • **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.

	Command	Purpose
Step 7	`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 8	`Router(config-access-point)# exit`	Exits access point configuration mode.

For information about the other access point configuration options, see the "Configuring Additional Real Access Point Options" section on page 7-20.

Configuring the IP Tunnel

When you configure a tunnel, you might consider using loopback interfaces as the tunnel endpoints instead of real 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 vrf forwarding vrf-name`	Associates a VRF instance with the interface.
Step 3	`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 4	`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 5	`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 Additional Real 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. Other access point configuration commands, if configured, will be ignored.

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-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.
Step 2	`Router(config-access-point)# aaa-group {authentication | accounting} server-group`	Specifies a default authentication, authorization, and accounting (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 an 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 3	`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 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)# 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 an interface to an external network on the GGSN. This is the default value. **Note** The default access-type is real. Therefore, you only need to configure this command if the APN needs to be a virtual access point.
Step 4	`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 5	`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. **Note** The **aggregate auto** command will not aggregate routes when using local IP address pools. **Note** This configuration applies to IPv4 PDP contexts.
Step 6	`Router(config-access-point)# anonymous user username [password]`	(Optional) Configures anonymous user access at an access point.
Step 7	`Router(config-access-point)# block-foreign-ms`	(Optional) Restricts GGSN access at a particular access point based on the mobile user's home PLMN.
Step 8	`Router(config-access-point)# cac-policy`	(Optional) Enables the maximum QoS policy function of the Call Admission Control (CAC) feature and applies a policy to an access point.
Step 9	`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. **Note** This configuration applies to IPv4 PDP contexts.
Step 10	`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. **Note** This configuration applies to IPv4 PDP contexts.
Step 11	`Router(config-access-point)# dns primary ip-address secondary ip-address`	(Optional) Specifies a primary (and backup) DNS to be sent in Create PDP Context responses at the access point. **Note** This configuration applies to IPv4 PDP contexts.

	Command	Purpose
Step 12	`Router(config-access-point)# gtp pdp-context single pdp-session [mandatory]`	(Optional) Configures the GGSN to delete the primary PDP context, and any associated secondary PDP contexts, of a *hanging* PDP session upon receiving a new create request from the same MS that shares the same IP address of the hanging PDP context. A hanging PDP context is a PDP context on the GGSN whose corresponding PDP context on the SGSN has already been deleted for some reason. When a hanging PDP session occurs and the **gtp pdp-context single pdp-session** command is not configured, if the same MS (on the same APN) sends a new Create PDP Context request that has a different NSAPI but has been assigned the same IP address used by the hanging PDP session, the GGSN rejects the new Create PDP Context request. When configure without the **mandatory** keyword specified, this feature applies only to those users for whom the Cisco vendor-specific attribute (VSA) "gtp-pdp-session=single-session" has been defined in their RADIUS user profile. To enable this feature and apply it to all users on an APN regardless of their RADIUS user profiles, specify the **mandatory** keyword option. **Note** If this feature is used with GTP load balancing, it might not function properly. **Note** This configuration applies to IPv4 PDP contexts.
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)# gtp pdp-context timeout idle interval [uplink]`	(Optional) Specifies the time, in seconds, that a GGSN allows a session to be idle at a particular access point before terminating the session.
Step 15	`Router(config-access-point)# gtp pdp-context timeout session interval [uplink]`	(Optional) Specifies the time, in seconds, that the GGSN allows a session to exist at any access point before terminating the session.

	Command	Purpose
Step 16	`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. **Note** To disable the sending of ICMP messages, ensure that the **no ip unreachable** interface configuration command has been configured on the virtual template interface. **Note** This configuration applies to IPv4 PDP contexts.
Step 17	`Router(config-access-point)# ip-address-pool` {**dhcp-proxy-client** | **radius-client** | **local** *pool-name* | **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. • **local**—Specifies that a local pool provides the IP address. This option requires that a local pool has been configured using the **ip local pool** global configuration command. • **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. **Note** This configuration applies to IPv4 PDP contexts.
Step 18	`Router(config-access-point)# ip probe path` *ip_address* **protocol udp** [**port** *port* **ttl** *ttl*]	(Optional) Enables the GGSN to send a probe packet to a specific destination for each PDP context that is successfully established on an APN. **Note** This configuration applies to IPv4 PDP contexts.
Step 19	`Router(config-access-point)# ipv6 ipv6-access-group` *ACL-name* [**up** | **down**]	(Optional) Applies an access-control list (ACL) configuration to uplink or downlink IPv6 payload packets.
Step 20	`Router(config-access-point)# ipv6 ipv6-address-pool` {**local** pool-name | **radius-client**}	(Optional) Configures a dynamic IPv6 prefix allocation method on an access-point.

	Command	Purpose
Step 21	`Router(config-access-point)# ipv6 base-vtemplate number`	(Optional) Specifies the virtual template interface, containing IPv6 routing advertisements (RA) parameters, for an APN to copy to create virtual sub-interfaces for IPv6 PDP contexts.
Step 22	`Router(config-access-point)# ipv6 dns primary ipv6-address [secondary ipv6-address]`	(Optional) Specifies the address of a primary (and backup) IPv6 DNS to be sent in IPv6 create PDP context responses on an access point.
Step 23	`Router(config-access-point)# ipv6 [enable | exclusive]`	(Optional) Configures an access point to allow both IPv6 and IPv4 PDP contexts, or to just allow IPv6 PDP contexts.
Step 24	`Router(config-access-point)# ipv6 redirect [all | intermobile] ipv6-address`	(Optional) Configures the GGSN to redirects IPv6 traffic to an external IPv6 device. The available options are: • **all—**Redirects all IPv6 traffic to an external IPv6 device for an APN. • **intermobile—**Redirects mobile-to-mobile IPv6 traffic to an external IPv6 device. • *ipv6-address*—IP address of the IPv6 external device to which you want to redirect IPv6 traffic.
Step 25	`Router(config-access-point)# ipv6 security verify source`	(Optional) Enables the GGSN to verify the IPv6 source address of an upstream TPDU against the address previously assigned to an MS.
Step 26	`Router(config-access-point)# msisdn suppression [value]`	(Optional) Specifies that the GGSN overrides the mobile station ISDN (MSISDN) number with a pre-configured value in its authentication requests to a RADIUS server.
Step 27	`Router(config-access-point)# nbns primary ip-address secondary ip-address`	(Optional) Specifies a primary (and backup) NetBIOS Name Service (NBNS) to be sent in the Create PDP Context responses to at the access-point. **Note** This configuration applies to IPv4 PDP contexts.
Step 28	`Router(config-access-point)# network-behind-mobile`	Enables an access point to support routing behind the mobile station (MS). **Note** This configuration applies to IPv4 PDP contexts.

	Command	Purpose
Step 29	`Router(config-access-point)# ppp-regeneration [max-session number | setup-time seconds | verify-domain | fix-domain | allow-duplicate]`	(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. • **verify-domain**—Configures the GGSN to verify the domain sent in the protocol configuration option (PCO) IE sent in a Create PDP Context request against the APN sent out by the user when PPP-regeneration is being used. If a mismatch occurs, the Create PDP Context request is rejected with the cause code "Service not supported." • **fix-domain**—Configures the GGSN to use the access point name as the domain name with which it initiates an L2TP tunnel to the user when PPP-regeneration is being used. The **ppp-regeneration fix-domain** and **ppp-regeneration verify-domain** command configurations are mutually exclusive. When the **ppp-regeneration fix-domain** command is configured, domain verification cannot be performed. • **allow-duplicate**—Configures the GGSN to not check for duplicate IP addresses for PPP regenerated PDP contexts. **Note** This configuration applies to IPv4 PDP contexts.
Step 30	`Router(config-access-point)# radius attribute acct-session-id charging-id`	(Optional) Specifies that the charging ID in the Acct-Session-ID (attribute 44) is included in access requests.
Step 31	`Router(config-access-point)# radius attribute nas-id format`	(Optional) Specifies that the GGSN sends the NAS-Identifier in access requests at the APN where *format* is a string sent in attribute 32 containing an IP address (%i), a host name (%h), and a domain name (%d).

	Command	Purpose
Step 32	`Router(config-access-point)# radius attribute suppress [imsi | qos | sgsn-address]`	(Optional) Specifies that the GGSN suppress the following in its authentication and accounting requests to a RADIUS server: • **imsi**—Suppresses the 3GPP-IMSI number. • **qos**—Suppresses the 3GPP-GPRS-Qos Profile. • **sgsn-address**—Suppresses the 3GPP-GPRS-SGSN-Address
Step 33	`Router(config-access-point)# radius attribute user-name msisdn`	(Optional) Specifies that the MSISDN is included in the User-Name (attribute 1) field in access requests.
Step 34	`Router(config-access-point) redirect all ip ip address`	(Optional) Configures the GGSN to redirect all traffic to an external device. **Note** This configuration applies to IPv4 PDP contexts.
Step 35	`Router(config-access-point) redirect intermobile ip ip address`	(Optional) Configures the GGSN to redirect mobile-to-mobile traffic to an external device. **Note** This configuration applies to IPv4 PDP contexts.
Step 36	`Router(config-access-point) security verify {source | destination}`	Specifies that the GGSN verify the source or destination address in Transport Protocol Data Units (TPDUs) received from a Gn interface. **Note** This configuration applies to IPv4 PDP contexts.
Step 37	`Router(config-access-point)# session idle-timer number`	(Optional) Specifies the time (between 1 and 168 hours) that the GGSN waits before purging idle mobile sessions for the current access point.
Step 38	`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 39	`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. **Note** This configuration applies to IPv4 PDP contexts.

Verifying the Real 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 7-28
- Verifying Reachability of the Network Through the Access Point, page 7-30

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** command 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.x
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service gprs ggsn
!
hostname ggsn
!
ip cef
!
...
!
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-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
!
gprs memory threshold 512
!
...
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:

```
GGSN# 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 Many factors can affect whether 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, use the **ping** command from a handset, or from 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, use the IP address (rather than the 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** command to see detailed statistics including send and receive byte and packet counts.

Tip To find the Terminal Identifier (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 Virtual Access Points on the GGSN

This section includes the following topics:

For a sample configuration, see the "Virtual APN Configuration Example" section on page 7-53.

Overview of the Virtual Access Point Feature

GGSN Release 3.0 and later support 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 home location register (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 also extend to those servers.

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.

Note On the Cisco 7600 series router platform, identical virtual APN configurations must exist on each GGSN that is load-balanced by means of a virtual server.

Benefits of the Virtual APN Feature

The virtual APN feature provides the following benefits:

- Simplifies provisioning of APN information
- 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
- By setting the APN from the AAA server (pre-authentication-based virtual APN), operators can work with any APN from the handset, including the wildcard APN (*) because the target APN the user is not connected to is base on the user provisioning.

General Restrictions of the Virtual APN Feature

The virtual APN feature has the following restrictions:

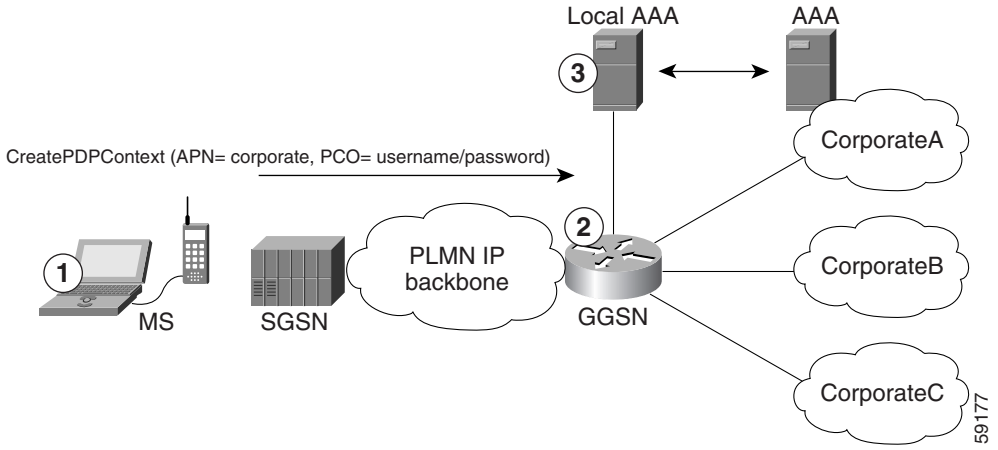
- CDRs do not include domain information because for virtual APNs, the domain information is removed from the username attribute. By default, the associated real APN name is used in CDRs and authentication requests to a virtual APN. However, the GGSN can be configured to send the virtual APN in CDRs using the **gprs charging cdr-option** command with the **apn virtual** keyword options specified.
- 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.

Domain-based Virtual Access Point s

By default, the GGSN determines the ultimate target network for a 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. Domain-based APN resolution is the default.

Figure 7-2 shows how the GGSN, by default, supports a Create PDP Context request from an MS processed through a virtual APN on the GGSN.

Figure 7-2 Default 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 request 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.

Pre-authentication-based Virtual Access Points

Cisco GGSN Release 6.0, Cisco IOS Release 12.3(14)YU and later, supports pre-authentication-based virtual access points.

The pre-authentication-based virtual APN feature utilizes AAA servers to provide dynamic, per-user mapping of a virtual APN to a target (real) APN.

When the **pre-authenticate** keyword option is specified when configuring a virtual APN, a pre-authentication phase is applied to Create PDP Context requests received that include a virtual APN in the APN information element.

Pre-authentication-based virtual APN requires that the AAA server be configured to provision user profiles to include the target APN. The AAA maps a user to the target using user identifications such as the IMSI, user name, or MSISDN, etc. Additionally, the target APN must be locally configured on the GGSN.

The following is the typical call flow with regard to external AAA servers when a virtual APN is involve:

1. The GGSN receives a Create PDP Context Request that includes a virtual APN. It locates the virtual APN and starts a pre-authentication phase for the PDP context by sending an Access-Request message to an AAA server.
2. The AAA server does a lookup based on the user identification (username, MSISDN, IMSI, etc.) included in the Access-Request message, and determines the target-APN for the user from the user profile. The target APN is returned as a Radius attribute in the Access-Accept message to the GGSN.
3. The GGSN checks for a locally-configured APN that matches the APN name in the target APN attribute in the Access-Accept message.
   - I f a match is found, the virtual APN is resolved and the Create PDP Context Request is redirected to the target APN and is further processed using the target APN (just as if the target APN was included in the original Create PDP Context request). If the real APN is non-transparent, another Access-Request is sent out. Typically, the AAA server should be different.
   - If a match is not found, the Create PDP Context Request is rejected.
   - If there is no target APN included in the RADIUS attribute in the access-accept message to the GGSN, or if the target APN is not locally configured, the Create PDP Context Request is rejected.
4. GGSN receives an access-accept from the AAA server for the second round of authentication.

Restrictions of the Pre-authentication-based Virtual APN Feature

In addition to the restrictions listed in the "General Restrictions of the Virtual APN Feature" section on page 7-32, when configuring pre-authentication-based virtual APN functionality, please note the following:

- If a user profile on the AAA server is configured to include a target APN, then the target APN should be a real APN, and it should be configured on the GGSN.
- An APN can only be configured for domain-based virtual APN functionality or pre-authentication-based APN functionality, not both.
- The target APN returned from AAA must be a real APN, and if more than one APN is returned, the first one is used and the rest ignored.
- Configure anonymous user access under the virtual APN (using the **anonymous user** access-point configuration command) to mobile stations (MS) to access without supplying the username and password (the GGSN uses the common password configured on the APN).
- At minimum, an AAA access-method must be configured under the virtual APN, or globally. If a method is not configured, the create PDP request will be rejected.

Virtual Access Point 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 required for connecting 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 7-35 (Required)
- Configuring Real Access Points on the GGSN, page 7-11 (Required)
  - PDN Access Configuration Task List, page 7-12
  - VPN Access Using VRF Configuration Task Lists, page 7-13
- Configuring Other GPRS/UMTS Network Entities With the Virtual APN, page 7-36 (Optional)

For a sample configuration, see the "Virtual APN Configuration Example" section on page 7-53.

Configuring Virtual Access Points on the GGSN

Use virtual access point types to consolidate access to multiple real target networks on the GGSN. Because the GGSN always uses real access points to reach an external network, 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 7-36.

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.
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 [pre-authenticate [default-apn apn-name]]`	Specifies an APN type that is not associated with any specific physical target network on the GGSN. Optionally, can be configured to be dynamically mapped, per user, to a target (default) APN. The default access type is real.

Note Even though the Cisco IOS software allows you to configure additional access point options on a virtual access point, none of those access point options will apply 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 for properly configuring other GPRS/UMTS network entities to support the virtual APN implementation.

The following GPRS/UMTS network entities might also require provisioning for proper implementation of 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 Access Point 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 7-36
- Verifying Reachability of the Network Through the Virtual Access Point, page 7-40

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** command 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:

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

Current configuration : 3521 bytes
!
version 12.x
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
!
ip cef
!
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
!
...
!
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
!
! Configure a domain-based 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
   !
! Configure a pre-authentication-based virtual access point called virtual-apn-all
!
 access-point 5
   access-point-name virtual-apn-all
   access-mode non-transparent
```

```
   access-type virtual pre-authenticate default-apn a1b1c1.com
   anonymous user anyone 1z1z1z
   radius attribute user-name msisdn
   exit
!
gprs maximum-pdp-context-allowed 90000
gprs gtp path-echo-interval 0
gprs default charging-gateway 10.15.15.1
!
gprs memory threshold 512
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
```

The following output shows information about a pre-authentication-based virtual access point that is configured to be dynamically mapped to a default APN named a1b1c1.com:

```
GGSN# show gprs access-point 5
    apn_index 1          apn_name = corporate
    apn_mode: non-transparent
    apn-type: Virtual pre-authenticate default-apn a1b1c1.com
    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 7-30.

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 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 Dynamic Addressing on the GGSN" chapter. For more information about configuring RADIUS on the GGSN, see the "Configuring Security on the GGSN" chapter.

Blocking Access to the GGSN by Foreign Mobile Stations

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

Overview of Blocking Foreign Mobile Stations

The GGSN allows you to block access by mobile stations that are outside of the PLMN. When you enable blocking of foreign mobile stations, the GGSN determines whether 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 whenever the GGSN receives a Create PDP Context request, 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.

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 its home PLMN.

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

- Configuring the MCC and MNC Values, page 7-41 (Required)
- Enabling Blocking of Foreign Mobile Stations on the GGSN, page 7-42 (Required)
- Verifying the Blocking of Foreign Mobile Stations Configuration, page 7-42

Configuring the MCC and MNC Values

The MCC and MNC together identify a public land mobile network (PLMN). The values that you configure using the **gprs mcc mnc** command without the **trusted** keyword option specified, are those of the home PLMN ID, which is the PLMN to which the GGSN belongs.

Only one home PLMN can be defined for a GGSN at a time. The GGSN compares the IMSI in Create PDP Context requests with the values configured using this command to determine if a request is from a foreign MS.

You can also configure up to 5 *trusted* PLMNs by specifying the **trusted** keyword when issuing the **gprs mcc mnc** command. A Create PDP Context request from an MS in a trusted PLMN is treated the same as a Create PDP Context request from an MS in the home PLMN.

To configure the MCC and MNC values that the GGSN uses to determine whether 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* `[trusted]`	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. Optionally, use the **trusted** keyword to define up to 5 trusted PLMNs. **Note** The Create PDP Context requests from a trusted PLMN are treated the same as those from the home PLMN.

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 before you can enable the GGSN to create CDRs for roamers.

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 access-point configuration mode:

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

Note The MCC and MNC values that are 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 to 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 7-42
- Verifying the MCC and MNC Configuration on the GGSN, page 7-43

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:

```
GGSN# 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 plmn** 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) and four trusted PLMNs have been configured:

```
GGSN# show gprs plmn
Home PLMN
  MCC = 302  MNC = 678
Trusted PLMN
  MCC = 346  MNC = 123
  MCC = 234  MNC = 67
  MCC = 123  MNC = 45
  MCC = 100  MNC = 35
```

Controlling Access to the GGSN by MSs with Duplicate IP Addresses

An MS cannot 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 Create 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.

Note On the Cisco 7600 series router platform, identical configurations must exist on each GGSN that is load-balanced by means of a virtual server.

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.

Configuring Routing Behind the Mobile Station on an APN

The routing behind the MS feature enables the routing of packets to IPv4 addresses that do not belong to the PDP context (the MS), but exist behind it. The network address of the destination can be different than the MS address.

Before enabling routing behind the MS, the following requirements must be met:

- The MS must use RADIUS for authentication and authorization.
- At minimum, one Framed-Route, attribute 22 as defined in Internet Engineering Task Force (IETF) standard RFC 2865, must be configured in the RADIUS server for each MS that wants to use this feature.

  When configured, the Framed-Route attribute is automatically downloaded to the GGSN during the authentication and authorization phase of the PDP context creation. If routing behind the MS is not enabled, the GGSN ignores the Framed-Route attribute. If multiple Framed-Route attributes have been configured for an MS, the GGSN uses the first attribute configured. When the MS session is no longer active, the route is deleted.
- For PPP Regen or PPP with L2TP sessions, the Framed-Route attribute must be configured in the RADIUS server of the LNS.
- For PPP Regen sessions, if the **security verify source** command is configure, the Framed-Route attribute must also be configured in the user profile in the GGSN RADIUS server.

Enabling Routing Behind the Mobile Station

To enable routing behind an MS, use the following command in access-point configuration mode:

Command	Purpose
`Router(config-access-point)# network-behind-mobile`	Enables an access point to support routing behind an MS.

Note The network-behind-mobile command applies to IPv4 PDP contexts.

Use the **show ip route** privilege EXEC command to view the current state of the routing table. To display a list of currently active mobile sessions, use the **show pdp** command.

Note Packets routed behind the MS share the same 3GPP QoS settings of the MS.

Verifying the Routing Behind the Mobile Station Configuration

To verify the routing behind the mobile station configuration, use the following **show** commands.

Step 1 From privilege EXEC mode, use the **show gprs gtp pdp-context tid** and **show ip route** commands to view the framed route and the static route added for the framed route that uses the IP address of the PDP context as the gateway address:

```
GGSN#show gprs gtp pdp-context tid 1234567809000010
TID              MS Addr                Source  SGSN Addr       APN
1234567809000010 83.83.0.1              Static  2.1.1.1         ippdp1

    current time :Feb 09 2004 12:52:49
    user_name (IMSI):214365879000000    MS address:83.83.0.1
    MS International PSTN/ISDN Number (MSISDN):123456789
    sgsn_addr_signal:2.1.1.1            sgsn_addr_data:  2.1.1.1
    control teid local: 0x637F00EC
    control teid remote:0x01204611
    data teid local:    0x637DFF04
    data teid remote:   0x01204612
    primary pdp:Y           nsapi:1
    signal_sequence: 11                 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:  Feb 09 2004 12:50:41
    last_access_time: Feb 09 2004 12:50:41
    mnrgflag:         0                tos mask map:00
    gtp pdp idle time:72
    gprs qos_req:000000               canonical Qos class(reg.):03
gprs qos_neg:000000               canonical Qos class(neg.):03
effective bandwidth:0.0
    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             out-sequence pkt:0
    charging_id:        736730069
    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
    Framed_route 5.5.5.0 mask 255.255.255.0
GGSN#
GGSN#show ip route
Codes:C - connected, S - static, 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
       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
C    2.0.0.0/8 is directly connected, FastEthernet6/0
     5.0.0.0/24 is subnetted, 1 subnets
U       5.5.5.0 [1/0] via 83.83.0.1
     83.0.0.0/32 is subnetted, 1 subnets
U       83.83.0.1 [1/0] via 0.0.0.0, Virtual-Access2
```

```
     8.0.0.0/32 is subnetted, 1 subnets
C       8.8.0.1 is directly connected, Loopback0
GGSN#
GGSN#show ip route vrf vpn4

Routing Table:vpn4
Codes:C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, 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

     80.0.0.0/16 is subnetted, 1 subnets
C       80.1.0.0 is directly connected, FastEthernet3/0
     5.0.0.0/24 is subnetted, 1 subnets
U       5.5.5.0 [1/0] via 123.123.123.123
     123.0.0.0/32 is subnetted, 1 subnets
U       123.123.123.123 [1/0] via 0.0.0.0, Virtual-Access9
GGSN#
```

Step 2 From privilege EXEC mode, use the show gprs gtp statistics command to view network-behind-mobile-station statistics (displayed in bold in the following example):

```
GGSN#show gprs gtp statistics
GPRS GTP Statistics:
  version_not_support        0            msg_too_short              0
  unknown_msg                0            unexpected_sig_msg         0
  unexpected_data_msg        0            unsupported_comp_exthdr    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    2            tft_semantic_error         0
  tft_syntactic_error        0            pkt_ftr_semantic_error     0
  pkt_ftr_syntactic_error    0            non_existent               0
  path_failure               0            total_dropped              0
  signalling_msg_dropped     0            data_msg_dropped           0
  no_resource                0            get_pak_buffer_failure     0
  rcv_signalling_msg         7            snd_signalling_msg         7
  rcv_pdu_msg                0            snd_pdu_msg                0
  rcv_pdu_bytes              0            snd_pdu_bytes              0
  total created_pdp          3            total deleted_pdp          2
  total created_ppp_pdp      0            total deleted_ppp_pdp      0
  ppp_regen_pending          0            ppp_regen_pending_peak     0
  ppp_regen_total_drop       0            ppp_regen_no_resource      0
  ntwk_init_pdp_act_rej      0            total ntwkInit created pdp 0
GPRS Network behind mobile Statistics:
  network_behind_ms APNs      1          total_download_route       5
  save_download_route_fail    0          insert_download_route_fail 2
  total_insert_download_route 3
```

Configuring Proxy-CSCF Discovery Support on an APN

The GGSN can be configured to return a list of preconfigured Proxy Call Session Control Function (P-CSCF) server addresses for an APN when it receives a Create PDP Context Request that contains a "P-CSCF Address Request" field in the PCO.

The MS sets the P-CSCF Address Request field of the PCO in the Activate PDP Context Request. This request is forwarded to the GGSN in the Create PDP Context Request from the SGSN. Upon receiving, the GGSN returns in the "P-CSCF Address" field of the PCO, all the P-CSCF addresses configured.

If a Create PDP Context Request does not contain the P-CSCF address request field in the PCO, or if no P-CSCF addresses are preconfigured, the Create PDP Context Response will not return any P-CSCF addresses. An error message will not be generated and the Create PDP Context Request will be processed.

Note The order of the addresses returned in the "P-CSCF Address Field" of the PCO is the same as the order in which they are defined in the P-CSCF server group and the groups are associated with the APN.

To enable the P-CSCF Discovery support on an APN, perform the following tasks:

- Creating P-CSCF Server Groups on the GGSN, page 7-47
- Specifying a P-CSCF Server Groups on an APN, page 7-47

Creating P-CSCF Server Groups on the GGSN

Up to 10 P-CSCF servers can be defined in a P-CSCF server group.

Both IPv6 and IPv4 P-CSCF servers can be defined in a server group. The PDP type dictates to which server the IP addresses are sent.

To configure a P-CSCF server group on the GGSN, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gprs pcscf group-name`	Configures a P-CSCF server group on the GGSN and enters P-CSCF group configuration mode.
Step 2	`Router(config-pcscf-group)# server [ipv6] ip-address`	Defines an IPv4 P-CSCF server by IP address. Optionally, specify the **ipv6** keyword option to define an IPv6 P-CSCF server in a P-CSCF server group.

Specifying a P-CSCF Server Groups on an APN

Before specifying a P-CSCF group on an APN, the group must be configured globally using the **gprs pcscf** global configuration command.

Note Only one P-CSCF group can be defined per APN.

To specify a P-CSCF server group for an APN, use the following command while in access point configuration mode:

Command	Purpose
`Router(config-access-point)# pcscf group-name`	Specifies a P-CSCF server group to be used for P-CSCF discovery by an APN.

Verifying the P-CSCF Discovery Configuration

Use the following show commands to verify the P-CSCF Discovery configuration:

Command	Purpose
`Router# show gprs pcscf`	Displays a summary of the P-CSCF server groups configured on the GGSN.
`Router# show gprs access-point [group-name]`	Displays a summary of the P-CSCF server group or groups configured on the GGSN.

Monitoring and Maintaining Access Points on the GGSN

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

Use the following privileged EXEC commands to monitor and maintain access points on the GGSN:

Command	Purpose
`Router# clear gprs access-point statistics {access-point-index | all}`	Clears statistics counters for a specific access point or for all access points on the GGSN.
`Router# clear gprs gtp pdp-context pdp-type [ipv6 | ipv4]`	clear all packet data protocol (PDP) contexts (mobile sessions) that are IP version 4 (IPv4) or IP version 6 (IPv6) PDPs
`Router# show gprs access-point {access-point-index | all}`	Displays information about access points on the GGSN.
`Router# show gprs access-point statistics {access-point-index | all}`	Displays data volume and PDP activation and deactivation statistics for access points on the GGSN.
`Router# show gprs access-point-name status`	Displays the number of active PDPs on an access point, and how many of those PDPs are IPv4 PDPs and how many are IPv6 PDPs.

Command	Purpose
`Router# clear gprs access-point statistics {access-point-index | all}`	Clears statistics counters for a specific access point or for all access points on the GGSN.
`Router# clear gprs gtp pdp-context pdp-type [ipv6 | ipv4]`	clear all packet data protocol (PDP) contexts (mobile sessions) that are IP version 4 (IPv4) or IP version 6 (IPv6) PDPs
`Router# show gprs access-point {access-point-index | all}`	Displays information about access points on the GGSN.
`Router# show gprs gtp pdp-context {tid tunnel_id [service [all | id id_string]] | ms-address ip_address [access-point access-point-index] | imsi imsi [nsapi nsapi [tft]] | path ip-address [remote-port-num] | access-point access-point-index | pdp-type {ip [v6 | v4] | ppp} | qos-umts-class {background | conversational | interactive | streaming} | qos-precedence {low | normal | high} | qos-delay {class1 | class2 | class3 | classbesteffort} | version gtp-version} | msisdn [msisdn] | ms-ipv6-addr ipv6-address | all}`	Displays a list of the currently active PDP contexts (mobile sessions).
`Router# show gprs gtp statistics`	Displays the current GTP statistics for the gateway GGSN (such as IE, GTP signaling, and GTP PDU statistics).
`Router# show gprs gtp status`	Displays information about the current status of the GTP on the GGSN.

Configuration Examples

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

Static Route to SGSN Example

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

GGSN Configuration:

```
!
...
!
interface Loopback100
 description GPRS GTP V-TEMPLATE IP ADDRESS
 ip address 9.9.9.72 255.255.255.0
!
interface GigabitEthernet0/0.2
 description Ga/Gn Interface
 encapsulation dot1Q 101
 ip address 10.1.1.72 255.255.255.0
 no cdp enable
!
interface Virtual-Template1
 description GTP v-access
 ip unnumbered Loopback100
 encapsulation gtp
 gprs access-point-list gprs
!
ip route 40.1.2.1 255.255.255.255 10.1.1.1
ip route 40.1.3.10 255.255.255.255 10.1.1.1
ip route 40.2.2.1 255.255.255.255 10.1.1.1
ip route 40.2.3.10 255.255.255.255 10.1.1.1
!
...
!
```

Supervisor Engine Configuration

```
!
...
!
interface FastEthernet8/22
 no ip address
 switchport
 switchport access vlan 302
!
interface FastEthernet9/41
 no ip address
 switchport
 switchport access vlan 303
!
interface Vlan101
 description Vlan to GGSN for GA/GN
 ip address 10.1.1.1 255.255.255.0
!
interface Vlan302
 ip address 40.0.2.1 255.255.255.0
!
interface Vlan303
 ip address 40.0.3.1 255.255.255.0
!

ip route 9.9.9.72 255.255.255.255 10.1.1.72
```

```
ip route 9.9.9.73 255.255.255.255 10.1.1.73
ip route 9.9.9.74 255.255.255.255 10.1.1.74
ip route 9.9.9.75 255.255.255.255 10.1.1.75
ip route 9.9.9.76 255.255.255.255 10.1.1.76
ip route 40.1.2.1 255.255.255.255 40.0.2.11
ip route 40.1.3.10 255.255.255.255 40.0.3.10
ip route 40.2.2.1 255.255.255.255 40.0.2.11
ip route 40.2.3.10 255.255.255.255 40.0.3.10
!
...
!
```

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 examples show a partial configuration for two VPNs (vpn1 and vpn2) and their associated GRE tunnel configurations (Tunnel1 and Tunnel2).

GGSN Configuration

```
service gprs ggsn
!
hostname 7600-7-2
```

```
!
ip cef
!
ip vrf vpn1
 description GRE Tunnel 1
 rd 100:1
!
ip vrf vpn2
 description GRE Tunnel 3
 rd 101:1
!
interface Loopback1
 ip address 150.1.1.72 255.255.0.0
!
interface Loopback100
 description GPRS GTP V-TEMPLATE IP ADDRESS
 ip address 9.9.9.72 255.255.255.0
!
interface Tunnel1
 description VRF-GRE to PDN 7500(13) Fa0/1
 ip vrf forwarding vpn1
 ip address 50.50.52.72 255.255.255.0
 tunnel source 150.1.1.72
 tunnel destination 165.2.1.13
!
interface Tunnel2
 description VRF-GRE to PDN PDN x(12) Fa3/0
 ip vrf forwarding vpn2
 ip address 80.80.82.72 255.255.255.0
 tunnel source 150.1.1.72
 tunnel destination 167.2.1.12
!
interface GigabitEthernet0/0.1
 description Gi
 encapsulation dot1Q 100
 ip address 10.1.2.72 255.255.255.0
!
interface Virtual-Template1
 description GTP v-access
 ip unnumbered Loopback100
 encapsulation gtp
 gprs access-point-list gprs
!
ip local pool vpn1_pool 100.2.0.1 100.2.255.255 group vpn1
ip local pool vpn2_pool 100.2.0.1 100.2.255.255 group vpn2
ip route vrf vpn1 0.0.0.0 0.0.0.0 Tunnel1
ip route vrf vpn2 0.0.0.0 0.0.0.0 Tunnel2

gprs access-point-list gprs
  access-point 1
   access-point-name apn.vrf1.com
   access-mode non-transparent
   aaa-group authentication ipdbfms
   ip-address-pool local vpn1_pool
   vrf vpn1
   !
  access-point 2
   access-point-name apn.vrf2.com
   access-mode non-transparent
   aaa-group authentication ipdbfms
   ip-address-pool local vpn2_pool
   vrf vpn2
   !
```

Supervisor Engine Configuration

```
interface FastEthernet9/5
 no ip address
 switchport
 switchport access vlan 167
 no cdp enable
!
interface FastEthernet9/10
 no ip address
 switchport
 switchport access vlan 165
 no cdp enable
!
interface Vlan165
 ip address 165.1.1.1 255.255.0.0
!
interface Vlan167
 ip address 167.1.1.1 255.255.0.0
!
! provides route to tunnel endpoints on GGSNs
!
ip route 150.1.1.72 255.255.255.255 10.1.2.72
!
! routes to tunnel endpoints on PDN
!
ip route 165.2.0.0 255.255.0.0 165.1.1.13
ip route 167.2.0.0 255.255.0.0 167.1.1.12
```

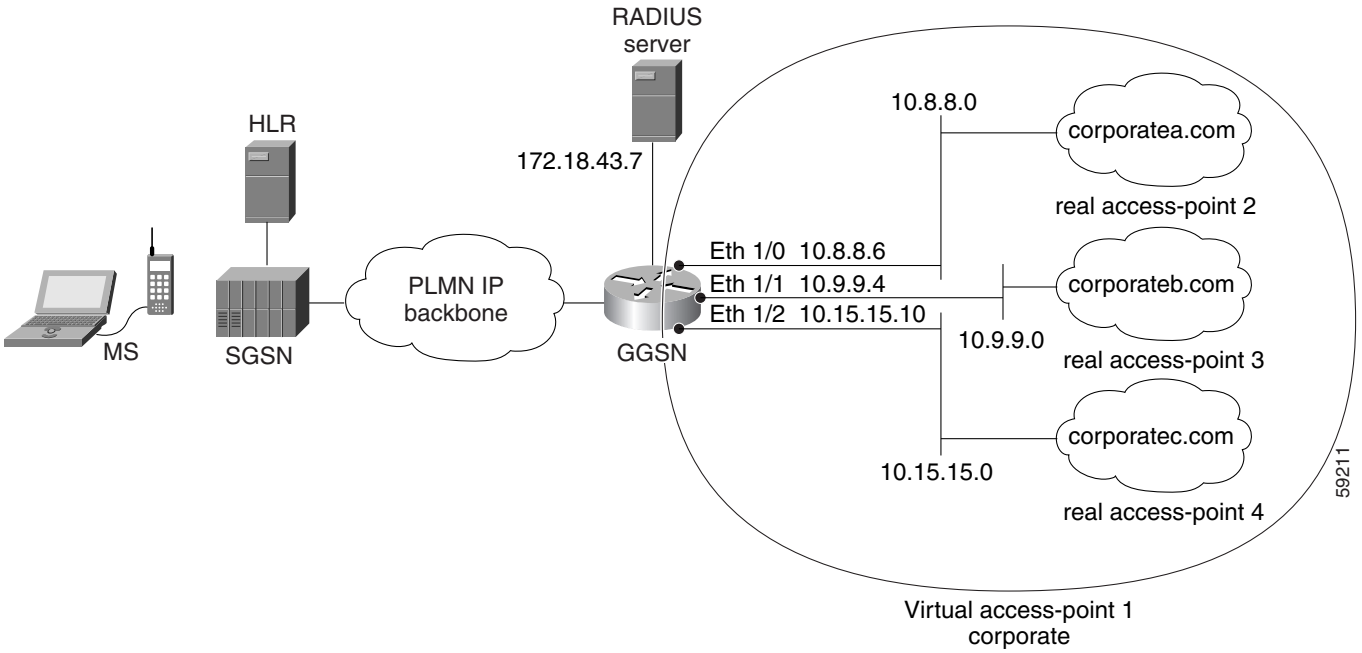
Virtual APN Configuration Example

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 7-3 Virtual APN Configuration Example



GGSN Configuration

```
!
version 12.x
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
!
ip cef
!
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 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 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 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
!
gprs memory threshold 512
!
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 in which access point 100 blocks access by foreign mobile stations:

```
!
version 12.x
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
!
ip cef
!
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
```

P-CSCF Discovery Configuration Example

The following example shows a partial configuration in which P-CSCF server groups have been configured on the GGSN and one is assigned to an access point:

```
!
version 12.x
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
!
ip cef
!
gprs pcscf groupA
server 172.10.1.1
server 10.11.1.2
server ipv6 2001:999::9
!
gprs pcscf groupB
server 172.20.2.1
server 10.21.2.2
gprs access-point-list gprs
!
access-point 100
access-point-name pcscf
pcscf groupA
!
```