



Configuring Load Balancing on the GGSN

This chapter describes how to configure a GGSN to support load balancing functions using the Cisco IOS software Server Load Balancing (SLB) feature. GGSN SLB provides increased reliability and availability when you are using multiple Cisco Systems GGSNs or non-Cisco GGSNs in your GPRS/UMTS network.

For a complete description of the GGSN commands in this chapter, refer to the *Cisco IOS Mobile Wireless Command Reference*. For a complete description of the other Cisco IOS SLB commands in this chapter, refer to the *IOS Server Load Balancing, 12.1(13)E3* documentation located at Cisco.com at the following URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121limit/121e/121e13/index.htm>

To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

This chapter includes the following sections:

- [Overview of Load Balancing on the GGSN, page 12-1](#)
- [Configuring GTP Load Balancing, page 12-6](#)
- [Monitoring and Maintaining the IOS SLB Feature, page 12-16](#)
- [Configuration Examples, page 12-17](#)



Note

You can also use the GTP Director Module (GDM) to obtain simple, round-robin load balancing in environments where non-Cisco GGSNs are found. For more information about GDM and load balancing, see the *GTP Director Module* section of this book.

Overview of Load Balancing on the GGSN

This section provides an overview of the Cisco IOS SLB feature and load balancing support on the GGSN. It includes the following sections:

- [Overview of Cisco IOS SLB, page 12-2](#)
- [GGSN GTP Load Balancing Support, page 12-2](#)
- [Configuration Guidelines, page 12-8](#)
- [Restrictions, page 12-6](#)

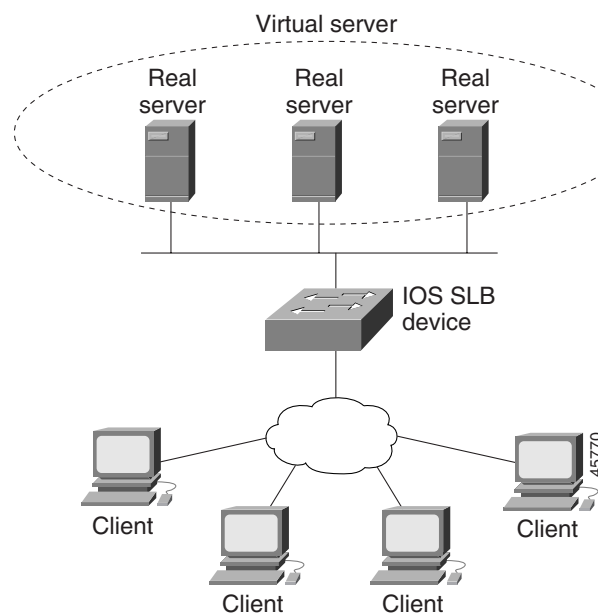
Overview of Cisco IOS SLB

The Cisco SLB feature is an IOS-based solution that provides IP server load balancing. Using the Cisco IOS SLB feature, you can define a *virtual server* that represents a group of *real servers* in a cluster of network servers known as a *server farm*. In this environment, the clients connect to the IP address of the virtual server. When a client initiates a connection to the virtual server, the Cisco IOS SLB feature chooses a real server for the connection based on a configured *load-balancing algorithm*.

The Cisco IOS SLB feature also provides firewall load balancing, which balances flows across a group of *firewalls* called a *firewall farm*.

Figure 12-1 illustrates a logical view of a simple Cisco IOS SLB network.

Figure 12-1 Logical View of IOS SLB



GGSN GTP Load Balancing Support

IOS SLB provides GGSN GTP load balancing and increased reliability and availability for the GGSN. GGSN GTP load balancing supports a subset of the overall server load balancing functions that are available in the Cisco IOS SLB feature. Therefore, the full scope of Cisco IOS SLB functions are not applicable to the GPRS/UMTS environment. For more information about unsupported functions, see the “Restrictions” section on page 12-6.

When configuring GTP load balancing, a pool of GGSNs is configured as a server farm in IOS SLB. These are the GGSNs across which you want to load balance GTP sessions. A virtual server instance is configured in IOS SLB to load balance GTP sessions across the GGSN farm. This virtual server is associated with the server farm that you configured in IOS SLB.

IOS SLB supports two types of GTP load balancing:

- [GTP Load Balancing without GTP Cause Code Inspection, page 12-3](#)
- [GTP Load Balancing with GTP Cause Code Inspection, page 12-3](#)

GTP Load Balancing without GTP Cause Code Inspection

GTP load balancing *without* GTP cause code inspection enabled is recommended for Cisco GGSNs. It has the following characteristics:

- Can operate in dispatched mode or in directed server NAT mode, but not in directed client NAT mode. In dispatched mode, the GGSNs must be Layer 2-adjacent to the IOS SLB device.
- Does not support stateful backup.
- Delivers tunnel creation messages destined to the virtual GGSN IP address to one of the real GGSNs, using the weighted round robin load-balancing algorithm. See the [“Weighted Round Robin” section on page 12-4](#) for more information about this algorithm.
- Requires DFP to account for secondary PDP contexts in GTP v1.

GTP Load Balancing with GTP Cause Code Inspection

GTP load balancing *with* GTP cause code inspection enabled allows IOS SLB to monitor all PDP context signaling flows to and from GGSN server farms. This enables IOS SLB to monitor GTP failure cause codes, detecting system-level problems in both Cisco and non-Cisco GGSNs.

[Table 1](#) lists the PDP create response cause codes and the corresponding actions taken by IOS SLB:

Table 1 PDP Create Response Cause Codes and Corresponding IOS SLB Actions

Cause Code	IOS SLB Action
Request Accepted	Establish session
No Resource Available	Fail current real, reassign session, drop the response
All dynamic addresses are occupied	Fail current real, reassign session, drop the response
No memory is available	Fail current real, reassign session, drop the response
System Failure	Fail current real, reassign session, drop the response
Missing or Unknown APN	Forward the response
Unknown PDP Address or PDP type	Forward the response
User Authentication Failed	Forward the response
Semantic error in TFT operation	Forward the response
Syntactic error in TFT operation	Forward the response
Semantic error in packet filter	Forward the response
Syntactic error in packet filter	Forward the response
Mandatory IE incorrect	Forward the response
Mandatory IE missing	Forward the response
Optional IE incorrect	Forward the response
Invalid message format	Forward the response
Version not supported	Forward the response

GTP load balancing *with* GTP cause code inspection enabled has the following characteristics:

- Must operate in directed server NAT mode.
- Assigns PDP context creates from a specific IMSI to the same GGSN.

- Supports stateful backup.
- Tracks the number of open PDP contexts for each GGSN, which enables GGSN server farms to use the weighted least connections (**leastconns**) algorithm for GTP load balancing. See the “[Weighted Least Connections](#)” section on page 12-4 for more information about this algorithm.
- Enables IOS SLB to deny access to a virtual GGSN if the carrier code of the requesting International Mobile Subscriber ID (IMSI) does not match a specified value.
- Enables IOS SLB to support secondary PDP contexts, even without DFP.

Weighted Round Robin

The weighted round robin algorithm specifies that the real server used for a new connection to the virtual server is chosen from the server farm in a circular fashion. Each real server is assigned a weight, n , that represents its capacity to handle connections, as compared to the other real servers associated with the virtual server. That is, new connections are assigned to a given real server n times before the next real server in the server farm is chosen.

For example, assume a server farm comprised of real server ServerA with $n = 3$, ServerB with $n = 1$, and ServerC with $n = 2$. The first three connections to the virtual server are assigned to ServerA, the fourth connection to ServerB, and the fifth and sixth connections to ServerC.



Note

Assigning a weight of $n=1$ to all of the servers in the server farm configures the IOS SLB device to use a simple round robin algorithm.

GTP load balancing *without* GTP cause code inspection enabled requires the weighted round robin algorithm. A server farm that uses weighted least connections can be bound to a virtual server providing GTP load balancing without GTP cause code inspection enabled, but you cannot place the virtual server **INSERVICE**. If you try to do so, IOS SLB issues an error message.

Weighted Least Connections

When GTP cause code inspection is enabled, GTP load balancing supports the Cisco IOS SLB weighted least connections algorithm.

The weighted least connections algorithm specifies that the next real server chosen from a server farm for a new connection to the virtual server is the server with the fewest active connections. Each real server is assigned a weight for this algorithm, also. When weights are assigned, the server with the fewest connections is based on the number of active connections on each server, and on the relative capacity of each server. The capacity of a given real server is calculated as the assigned weight of that server divided by the sum of the assigned weights of all of the real servers associated with that virtual server, or $n_1/(n_1+n_2+n_3\dots)$.

For example, assume a server farm comprised of real server ServerA with $n = 3$, ServerB with $n = 1$, and ServerC with $n = 2$. ServerA would have a calculated capacity of $3/(3+1+2)$, or half of all active connections on the virtual server, ServerB one-sixth of all active connections, and ServerC one-third of all active connections. At any point in time, the next connection to the virtual server would be assigned to the real server whose number of active connections is farthest below its calculated capacity.

**Note**

Assigning a weight of $n=1$ to all of the servers in the server farm configures the IOS SLB device to use a simple least-connection algorithm.

GTP load balancing *without* GTP cause code inspection enabled *does not* support the weighted least connections algorithm.

GTP load balancing *with* GTP cause code inspection *does* support the weighted least connections algorithm.

Dynamic Feedback Protocol for IOS SLB

In GTP load balancing, IOS SLB knows when a PDP context is established, but it does not know when PDP contexts are cleared, and therefore it cannot know the number of open PDP contexts for each GGSN. Use the IOS SLB Dynamic Feedback Protocol (DFP) to calculate GPRS/UMTS load-balancing weights dynamically.

With IOS SLB DFP support, a *DFP manager* in a load-balancing environment can initiate a TCP connection with a *DFP agent*. Thereafter, the DFP agent collects status information from one or more real host servers, converts the information to relative weights, and reports the weights to the DFP manager. The DFP manager factors in the weights when load balancing the real servers. In addition to reporting at user-defined intervals, the DFP agent sends an early report if there is a sudden change in a real server's status.

The weights calculated by DFP override the static weights you define using the **weight (server farm)** command. If DFP is removed from the network, IOS SLB reverts to the static weights.

You can define IOS SLB as a DFP manager, as a DFP agent for another DFP manager (such as DistributedDirector), or as both at the same time. In such a configuration, IOS SLB sends periodic reports to DistributedDirector, which uses the information to choose the best server farm for each new connection request. IOS SLB then uses the same information to choose the best real server within the chosen server farm.

DFP also supports the use of multiple DFP agents from different client subsystems (such as IOS SLB and GPRS/UMTS) at the same time.

In GTP load balancing, you can define IOS SLB as a DFP manager and define a DFP agent on each GGSN in the server farm, and the DFP agent can report the weights of the GGSNs. The DFP agents calculate the weight of each GGSN based on CPU utilization, processor memory, and the maximum number of PDP contexts (mobile sessions) that can be activated for each GGSN.

The weight for each GGSN is primarily based on the ratio of existing PDP contexts on the GGSN and the maximum number of allowed PDP contexts. CPU and memory utilization become part of the weight calculation only after the utilization exceeds 85%. Because the maximum number of allowed PDP contexts is considered to be the GGSNs maximum load, you should carefully consider the value that you configure in the **gprs maximum-pdp-context-allowed** command, which defaults to 10000 PDP contexts.

Restrictions

The following restrictions apply when configuring GTP load balancing:

- For GTP load balancing without GTP cause code inspection enabled:
 - Operates in either dispatched mode or directed server NAT mode only.
 - Cannot load balance network-initiated PDP context requests.
 - Does not support the following Cisco IOS SLB functions:
 - Bind IDs
 - Client-assigned load balancing
 - Slow Start
 - Stateful backup
 - Sticky connections
 - Weighted least connections load-balancing algorithm.
- For GTP load balancing *with* GTP cause code inspection enabled:
 - Operates in directed server NAT mode only.
 - Cannot load-balance network-initiated PDP context requests.
 - Requires either the SGSN or the GGSN to echo its peer.
 - Inbound and outbound traffic should be routed via IOS SLB.
 - Does not support the following IOS SLB functions:
 - Bind IDs
 - Client-assigned load balancing
 - Slow Start
 - Sticky connections

Configuring GTP Load Balancing

This section includes the following topics:

- [GTP Load Balancing Configuration Task List, page 12-7](#)
- [Configuration Guidelines, page 12-8](#)
- [Verifying the IOS SLB Configuration, page 12-14](#)

GTP Load Balancing Configuration Task List

This section lists the tasks used to configure GTP load balancing. Detailed configuration information is contained in the referenced sections of this or other documents. Required and optional tasks are indicated.

- On the IOS SLB, complete the following tasks:
 - [Configuring a Server Farm and Real Server, page 12-8](#) (Required)
 - [Configuring a Virtual Server, page 12-10](#) (Required)
 - Configuring the virtual IP address as a loopback on each of the GGSNs in the server (Required if using dispatched mode)

This step is required only if you are using dispatched mode *without* GTP cause code inspection enabled. See the “Configuring a Loopback Interface” section in the *Cisco IOS Interface Configuration Guide* for more information.
 - [Configuring a GSN Idle Timer, page 12-12](#) (Optional if GTP cause code inspection is enabled.)
 - [Configuring DFP, page 12-13](#) (Optional but recommended)
- On the GGSN, complete the following tasks:
 - If using DFP:
 - [Configuring the Maximum DFP Weight for a GGSN, page 12-13](#) (Optional)
 - [Configuring the Maximum Number of PDP Contexts for a GGSN, page 12-13](#) (Optional)
 - [Identifying the GGSN Virtual Server to CEF, page 12-14](#) (Required if using CEF)
 - Routing each GGSN to each associated SGSN (Required)

The route can be static or dynamic but the GGSN needs to be able to reach the SGSN. For more information, see the “[Configuring a Route to the SGSN](#)” section on [page 6-3](#) of the “[Configuring Network Access to the GGSN](#)” chapter.
- On the SGSN, route each SGSN to the virtual templates on each associated GGSN, and to the GGSN load-balancing virtual server (Required)

See the configuration guide for your SGSN for more details.

Configuration Guidelines

When configuring the network shared by IOS SLB and the GGSNs, keep the following considerations in mind:



- Specify static routes (using **ip route** commands) and real server IP addresses (using **real** commands) such that the Layer 2 information is correct and unambiguous.
- Choose subnets carefully, using one of the following methods:
 - Do not overlap virtual template address subnets.
 - Specify next hop addresses to real servers, not to interfaces on those servers.
- IOS SLB supports two types of GTP load balancing:
 - [GTP Load Balancing without GTP Cause Code Inspection, page 12-3](#)
 - [GTP Load Balancing with GTP Cause Code Inspection, page 12-3](#)
- IOS SLB supports both GTP v0 and GTP v1. Support for GTP enables IOS SLB to become “GTP aware,” extending IOS SLB’s knowledge into Layer 5.
- If you have enabled Cisco Express Forwarding (CEF) on a GGSN, you must identify the IP address of the GGSN virtual server to CEF. If you have *not* enabled CEF on the GGSN, do not perform this task. See the “[Identifying the GGSN Virtual Server to CEF](#)” section on [page 12-14](#) for more details.



Configuring a Server Farm and Real Server

When you configure the server farm and real server on the IOS SLB for GTP load balancing, use the following guidelines to ensure proper configuration:

- If GTP cause code inspection is not enabled, accept the default setting (the weighted round robin algorithm) for the **predictor** command.
If GTP cause code inspection is enabled, you can specify either the weighted round robin algorithm (**roundrobin**) or the weighted least connections (**leastconns**) algorithm.
- Specify the IP addresses (virtual template addresses, for Cisco GGSNs) of the real servers performing the GGSN function, using the **real** command.
- Specify a reassign threshold less than the GGSN’s N3-REQUESTS counter value using the **reassign** command.

To configure an IOS SLB server farm, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# ip slb serverfarm <i>serverfarm-name</i> Router(config-slb-sfarm)#	Adds a server farm definition to the IOS SLB configuration and enters server farm configuration mode.
Step 2	Router(config-slb-sfarm)# predictor [roundrobin leastconns]	Specifies the algorithm to be used to determine how a real server is selected.  Note In GTP load balancing without GTP cause code inspection enabled, you must accept the default setting (the weighted round robin algorithm). See the following sections for more details about each algorithm: <ul style="list-style-type: none"> • Weighted Round Robin, page 12-4 • Weighted Least Connections, page 12-4
Step 3	Router(config-slb-sfarm)# nat server	(Required if GTP cause code inspection is enabled) Configures NAT server address translation mode on the server farm.
Step 4	Router(config-slb-sfarm)# real <i>ip-address</i> [<i>port</i>]	Identifies a real GGSN as a member of a server farm using the IP address of the GGSN's virtual template interface, and enters real server configuration mode.
Step 5	Router(config-slb-real)# faildetect numconns <i>number-conns</i> [numclients <i>number-clients</i>]	(Optional) Specifies the number of consecutive connection failures and, optionally, the number of unique client connection failures, that constitute failure of the real server. In GTP load balancing, if there is only one SGSN in your environment, specify the numclients keyword with a value of 1.
Step 6	Router(config-slb-real)# maxconns <i>number-conns</i>	(Optional) Specifies the maximum number of active connections allowed on the real server at one time.  Note The impact of this command in GTP load balancing <i>without</i> GTP cause code inspection enabled is minimal because sessions are very short-lived.

	Command	Purpose
Step 7	Router(config-slb-real)# reassign <i>threshold</i>	<p>(Optional) Specifies the threshold of consecutive unacknowledged synchronizations or create PDP context requests that, if exceeded, result in an attempted connection to a different real server.</p> <p> Note In GTP load balancing, you must specify a reassign threshold less than the SGSN's N3-REQUESTS counter value.</p>
Step 8	Router(config-slb-real)# retry <i>retry-value</i>	<p>(Optional) Specifies the interval, in seconds, to wait between the detection of a server failure and the next attempt to connect to the failed server.</p>
Step 9	Router(config-slb-real)# weight <i>weighting-value</i>	<p>(Optional) Specifies the real server's workload capacity relative to other servers in the server farm.</p> <p> Note If you use DFP, the static weights you define using the weight (server farm) command are overridden by the weights calculated by DFP. If DFP is removed from the network, IOS SLB reverts to the static weights.</p>
Step 10	Router(config-slb-real)# inservice	Enables the real server for use by IOS SLB.

Configuring a Virtual Server




When you configure the virtual server on the IOS SLB for GTP load balancing, use the following guidelines to ensure proper configuration:

- Specify a virtual GGSN IP address as the virtual server, and use the **udp** keyword option.
- To load-balance GTP v1 sessions, specify port number **2123**, if the GGSNs and SGSNs are in compliance with the ETSI standard, or specify port number **0** or **any** to configure an all-port virtual server (that is, a virtual server that accepts flows destined for all ports).
- To load-balance GTP v0 sessions, specify port number **3386**, if the GGSNs and SGSNs are in compliance with the ETSI standard, or specify port number **0** or **any** to configure an all-port virtual server.
- To enable GTP load balancing *without* GTP cause code inspection, specify the **service gtp** keyword option.
- To enable GTP load balancing *with* GTP cause code inspection, specify the **service gtp-inspect** keyword option.

In GTP load balancing *without* GTP cause code inspection enabled, when you configure the idle timer using the **idle** command, specify an idle timer greater than the longest possible interval between PDP context requests on the SGSN.

To configure an IOS SLB virtual server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# ip slb vserver <i>virtual_server-name</i>	Identifies a virtual server and enters virtual server configuration mode.
Step 2	Router(config-slb-vserver)# virtual <i>ip-addr</i> [<i>netmask</i> [group]] { esp gre <i>protocol</i> } or Router(config-slb-vserver)# virtual <i>ip-addr</i> [<i>netmask</i> [group]] { tcp udp } [<i>port</i> any] [service <i>service</i>]	Specifies the virtual server IP address, type of connection, and optional TCP or UDP port number, IKE (ISAKMP) or WSP setting, and service coupling. Note For GTP load balancing: <ul style="list-style-type: none"> – Specify a virtual GGSN IP address as the virtual server, and specify the udp keyword option. – To load-balance GTP v1 sessions, specify port number 2123, if the GGSNs and SGSNs are in compliance with the ETSI standard, or specify port number 0 or any to configure an all-port virtual server (that is, a virtual server that accepts flows destined for all ports). – To load-balance GTP v0 sessions, specify port number 3386, if the GGSNs and SGSNs are in compliance with the ETSI standard, or specify port number 0 or any to configure an all-port virtual server. – To enable GTP load balancing <i>without</i> GTP cause code inspection, specify the service gtp keyword option. – To enable GTP load balancing <i>with</i> GTP cause code inspection, specify the service gtp-inspect keyword option.
Step 3	Router(config-slb-vserver)# serverfarm <i>primary-farm</i> [backup <i>backup-farm</i>]	Associates a real server farm with a virtual server, or configures a backup server farm.

	Command	Purpose
Step 4	Router(config-slb-vserver)# idle [gtp request] duration	<p>(Optional) Specifies the minimum amount of time IOS SLB maintains connection context in the absence of packet activity.</p> <p> Note In GTP load balancing <i>without</i> GTP cause code inspection enabled, specify an idle timer greater than the longest possible interval between PDP context requests on the SGSN.</p> <p>The gtp request keyword option is only supported in GTP load balancing with GTP cause code inspection enabled.</p>
Step 5	Router(config-slb-vserver)# inservice	Enables the virtual server for use by IOS SLB.
Step 6	Router(config-slb-vserver)# client {ip-address network-mask [exclude] gtp carrier-code [code]}	<p>(Optional) Specifies which clients are allowed to use the virtual server.</p> <p> Note GTP load balancing supports only the gtp carrier-code option, and only if GTP cause code inspection is enabled.</p>
Step 7	Router(config-slb-vserver)# replicate casa listen-ip remote-ip port [interval] [password [0 7] password timeout]	<p>(Optional) Configures a stateful backup of IOS SLB decision tables to a backup switch.</p> <p> Note GTP load balancing <i>without</i> GTP cause code inspection enabled does not support this command.</p>

Configuring a GSN Idle Timer

When GTP cause code inspection is enabled, you can configure the amount of time the IOS SLB will maintain sessions to and from an idle GGSN or SGSN.

To configure a GSN idle timer, enter the following command in global configuration mode on the IOS SLB:

Command	Purpose
Router(config)# ip slb timers gtp gsn duration	Changes the amount of time IOS SLB maintains sessions to and from an idle GGSN or SGSN.

Configuring DFP

You can define IOS SLB as a DFP manager, as a DFP agent for another DFP manager (such as DistributedDirector), or as both at the same time. Depending on your network configuration, you might enter the commands for configuring IOS SLB as a DFP manager and the commands for configuring IOS SLB as a DFP agent on the same device or on different devices.

To configure IOS SLB as a DFP manager, and to identify a DFP agent with which IOS SLB can initiate connections, use the following commands beginning in global configuration mode:

	Command	Description
Step 1	Router(config)# ip slb dfp [password [0 7] <i>password</i> [<i>timeout</i>]]	Configures DFP, supplies an optional password, and enters DFP configuration mode.
Step 2	Router(config-slb-dfp)# agent <i>ip_address port-number</i> [<i>timeout</i> [<i>retry_count</i> [<i>retry_interval</i>]]]	Identifies a DFP agent to which IOS SLB can connect.

To configure IOS SLB as a DFP agent, see the *DFP Agent Subsystem* feature module.

Configuring the Maximum DFP Weight for a GGSN

If you use DFP with GTP load balancing, each GGSN that acts as a DFP agent has a maximum weight that it can send to a DFP manager. For each GGSN, you can accept the default maximum weight, or you can specify a different maximum weight.

To specify the maximum weight for a GGSN, use the following command in global configuration mode on the GGSN:

Command	Purpose
Router(config)# gprs dfp max-weight [<i>max-weight-value</i>]	Specifies the maximum weight of a GGSN that is acting as a DFP agent.

Configuring the Maximum Number of PDP Contexts for a GGSN

If you use DFP with GTP load balancing, you must specify a maximum number of PDP contexts for each GGSN, using the **gprs maximum-pdp-context-allowed** command. *Do not* accept the default value of 10000 PDP contexts. A value of **45000** is recommended. Significantly lower values, including the default value of 10000, can impact performance in a GPRS/UMTS load-balancing environment.

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

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

For more information about the number of PDP contexts supported on a GGSN, see the [“Configuring the Maximum Number of PDP Contexts on the GGSN”](#) section on page 4-15 of the “Configuring GGSN GTP Services” chapter.

Identifying the GGSN Virtual Server to CEF

If you have enabled CEF on a GGSN and are using dispatched mode, you must identify the IP address of the GGSN virtual server to CEF. (This IP address is also a loopback address on the GGSN.)

If you have *not* enabled CEF on the GGSN and are not using dispatched mode, do not perform this task.

To identify the IP address of the GGSN virtual server to CEF, use the following command in global configuration mode on the GGSN:

Command	Purpose
Router(config)# gprs slb cef [virtual-server-address]	Specifies the IP address of the GGSN virtual server instance used by clients to connect to the server farm, for use by CEF. This command is required only if the GGSN is using CEF switching. Do not use this command if the GGSN is not using CEF switching.

Verifying the IOS SLB Configuration

This section describes how to verify the IOS SLB configuration. It includes the following topics:

- [Verifying the Virtual Server, page 12-14](#)
- [Verifying the Server Farm, page 12-15](#)
- [Verifying the Clients, page 12-15](#)
- [Verifying IOS SLB Connectivity, page 12-15](#)

Verifying the Virtual Server

The following **show ip slb vserver** command verifies the configuration of the virtual servers PUBLIC_HTTP and RESTRICTED_HTTP:

```
Router# show ip slb vserver

slb vserver      prot  virtual                state      conns
-----
PUBLIC_HTTP     TCP   10.0.0.1:80            OPERATIONAL  0
RESTRICTED_HTTP TCP   10.0.0.2:80            OPERATIONAL  0
Router#
```

Verifying the Server Farm

The following **show ip slb reals** command displays the status of server farms PUBLIC and RESTRICTED, the associated real servers, and their status:

```
Router# show ip slb real
```

real	farm name	weight	state	conns
10.1.1.1	PUBLIC	8	OPERATIONAL	0
10.1.1.2	PUBLIC	8	OPERATIONAL	0
10.1.1.3	PUBLIC	8	OPERATIONAL	0
10.1.1.20	RESTRICTED	8	OPERATIONAL	0
10.1.1.21	RESTRICTED	8	OPERATIONAL	0

```
Router#
```

The following **show ip slb serverfarm** command displays the configuration and status of server farms PUBLIC and RESTRICTED:

```
Router# show ip slb serverfarm
```

server farm	predictor	nat	reals	bind id
PUBLIC	ROUNDROBIN	none	3	0
RESTRICTED	ROUNDROBIN	none	2	0

```
Router#
```

Verifying the Clients

The following **show ip slb conns** command verifies the restricted client access and status:

```
Router# show ip slb conns
```

vserver	prot	client	real	state	nat
RESTRICTED_HTTP	TCP	10.4.4.0:80	10.1.1.20	CLOSING	none

```
Router#
```

The following **show ip slb conns** command displays detailed information about the restricted client access status:

```
Router# show ip slb conns client 10.4.4.0 detail
VSTEST_UDP, client = 10.4.4.0:80
  state = CLOSING, real = 10.1.1.20, nat = none
  v_ip = 10.0.0.2:80, TCP, service = NONE
  client_syms = 0, sticky = FALSE, flows attached = 0
Router#
```

Verifying IOS SLB Connectivity

To verify that the IOS SLB feature has been installed and is operating correctly, ping the real servers from the IOS SLB switch, then ping the virtual servers from the clients.

The following **show ip slb stats** command displays detailed information about the IOS SLB network status:

```
Router# show ip slb stats
Pkts via normal switching: 0
Pkts via special switching: 0
Pkts via slb routing: 0
Pkts Dropped: 0
```

```

Connections Created:          0
Connections Established:     0
Connections Destroyed:       0
Connections Reassigned:      0
Zombie Count:                0
Connections Reused:          0
Connection Flowcache Purges: 0
Failed Connection Allocs:    0
Failed Real Assignments:     0
RADIUS framed-ip Sticky Count:0
RADIUS username Sticky Count: 0

```

See the “[Monitoring and Maintaining the IOS SLB Feature](#)” section on page 12-16 for additional commands used to verify IOS SLB networks and connections.

Monitoring and Maintaining the IOS SLB Feature

To obtain and display runtime information about IOS SLB for the GGSN, use the following commands in privileged EXEC mode:

Command	Purpose
Router# show ip slb conns [vserver <i>virtual_server-name</i> client <i>ip-address</i> firewall <i>firewallfarm-name</i>] [detail]	Displays all connections handled by IOS SLB, or, optionally, only those connections associated with a particular virtual server or client.
Router# show ip slb dfp [agent <i>agent_ip_address</i> <i>port-number</i> manager <i>manager_ip_address</i> detail weights]	Displays information about DFP and DFP agents, and about the weights assigned to real servers.
Router# show ip slb gtp { gsn [<i>gsn-ip-address</i>] nsapi [<i>nsapi-key</i>] [detail]	Displays IOS SLB GTP information.
Router# show ip slb reals [sfarm <i>server-farm</i>] [detail]	Displays information about the real servers defined to IOS SLB.
Router# show ip slb replicate	Displays information about the IOS SLB replication configuration.
Router# show ip slb serverfarms [name <i>serverfarm-name</i>] [detail]	Displays information about the server farms defined to IOS SLB.
Router# show ip slb sessions [gtp gtp-inspect radius] [vserver <i>virtual-server</i>] [client <i>ip-addr netmask</i>] [detail]	Displays information about sessions handled by IOS SLB.
Router# show ip slb stats	Displays IOS SLB statistics.
Router# show ip slb vserver [name <i>virtual_server</i>] [redirect] [detail]	Displays information about the virtual servers defined to IOS SLB.

Configuration Examples

This section provides real-world examples of the GGSN IOS SLB examples. For a complete description of the GGSN commands in this section, see the Cisco IOS Mobile Wireless Command Reference. For a complete description of the IOS SLB commands in this section, see the *IOS Server Load Balancing*, 12.1(13)E3 documentation.

This section includes the following examples:

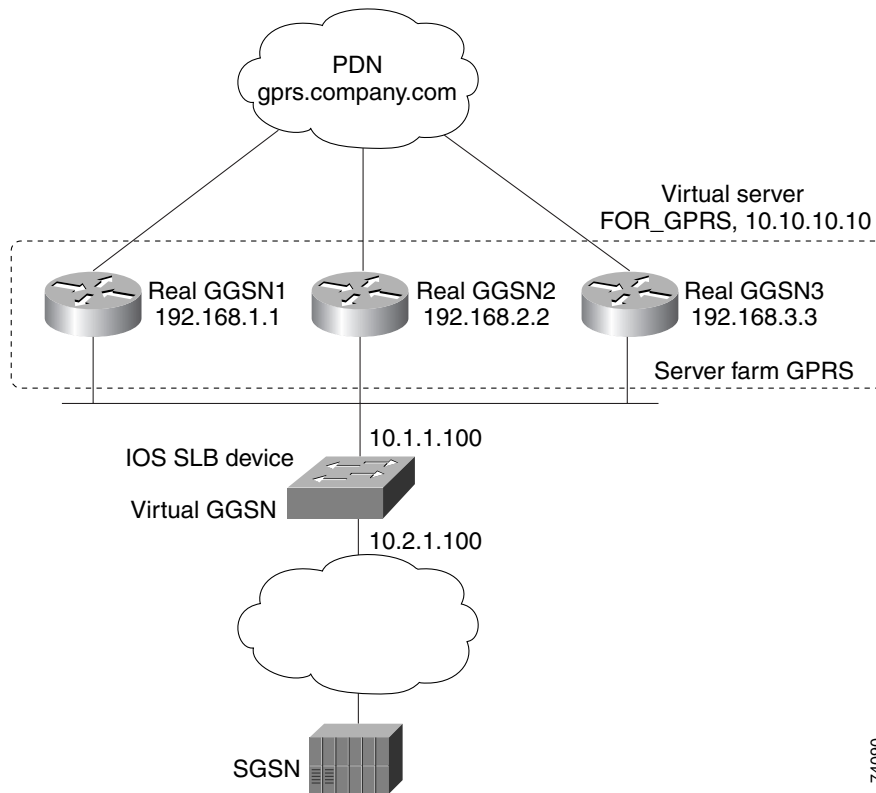
- [IOS SLB with GTP Load Balancing Configuration Example, page 12-17](#)
- [IOS SLB with GTP Load Balancing and NAT Example, page 12-22](#)
- [IOS SLB with GTP Load Balancing, NAT, and GTP Cause Code Inspection Example, page 12-25](#)

IOS SLB with GTP Load Balancing Configuration Example

[Figure 2](#) shows a typical GTP load-balancing configuration *without* GTP cause code inspection enabled. In this configuration:

- IOS SLB can balance GTP flows across multiple real GGSNs. The SGSN “sees” the real GGSNs as a single virtual GGSN. This configuration increases the flow-handling capability of the real GGSNs and increases the reliability and availability.
- The ip address of the SGSN is 10.111.111.111.
- The virtual template address of GGSN1 is 192.168.1.1.
- The virtual template address of GGSN2 is 192.168.2.2.
- The virtual template address of GGSN3 is 192.168.3.3.

Figure 2 IOS SLB with GTP Load Balancing



74090

Following are the configuration statements for the configuration shown in [Figure 2](#):

- [IOS SLB Configuration Statements, page 12-19](#)
- [GGSN1 Configuration Statements, page 12-19](#)
- [GGSN2 Configuration Statements, page 12-20](#)
- [GGSN3 Configuration Statements, page 12-21](#)

For more detailed GGSN configuration examples, see the *Cisco IOS Mobile Wireless Configuration Guide*.

IOS SLB Configuration Statements

```

hostname GTP_SLB
!
ip domain-name gprs.com
!
ip slb serverfarm GPRS
  real 192.168.1.1
    weight 1
    faildetect numconns 1 numclients 1
  inservice
!
  real 192.168.2.2
    weight 1
    faildetect numconns 1 numclients 1
  inservice
!
  real 192.168.3.3
    weight 1
    faildetect numconns 1 numclients 1
  inservice
!
ip slb vserver FOR_GPRS
  virtual 10.10.10.10 udp 3386 service gtp
  serverfarm GPRS
  inservice
!
ip slb dfp password Cookies 0
  agent 10.1.1.201 1111 30 0 10
  agent 10.1.1.202 1111 30 0 10
  agent 10.1.1.203 1111 30 0 10
!
interface FastEthernet1/0
  description TO SERVERFARM GPRS
  ip address 10.1.1.100 255.255.255.0
  no ip redirects
  duplex half
!
interface FastEthernet3/0
  description TO SGSN
  ip address 10.2.1.100 255.255.255.0
  no ip mroute-cache
  duplex half
!
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
ip route 192.168.1.1 255.255.255.255 10.1.1.201
ip route 192.168.2.2 255.255.255.255 10.1.1.202
ip route 192.168.3.3 255.255.255.255 10.1.1.203

```

GGSN1 Configuration Statements

```

service gprs ggsn
!
hostname GGSN1
!
ip dfp agent gprs
  port 1111
  password Cookies 0
  inservice
!
ip domain-name gprs.com
!

```

```

interface loopback 1
  description LOOPBACK SAME AS IOS SLB VSERVER ADDRESS
  ip address 10.10.10.10 255.255.255.255
  no ip route-cache
  no ip mroute-cache
!
interface FastEthernet1/0
  description TO SLB
  ip address 10.1.1.201 255.255.255.0
  ip directed-broadcast
  no ip mroute-cache
  duplex half
!
interface Virtual-Template1
  description GTP VIRTUAL TEMPLATE
  ip address 192.168.1.1 255.255.255.0
  encapsulation gtp
  gprs access-point-list gprs1
!
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
!
gprs access-point-list gprs1
  access-point 1
    access-point-name gprs.company.com
    access-mode non-transparent
    ip-address-pool dhcp-proxy-client
    dhcp-server 10.100.0.5 10.100.0.6
    dhcp-gateway-address 10.27.3.1
    exit
!
gprs maximum-pdp-context-allowed 45000
gprs qos map canonical-qos
gprs gtp path-echo-interval 0
gprs dfp max-weight 32
gprs slb cef 10.10.10.10

```

GGSN2 Configuration Statements

```

service gprs ggsn
!
hostname GGSN2
!
ip dfp agent gprs
  port 1111
  password Cookies 0
  inservice
!
ip domain-name gprs.com
!
interface loopback 1
  description LOOPBACK SAME AS IOS SLB VSERVER ADDRESS
  ip address 10.10.10.10 255.255.255.255
  no ip route-cache
  no ip mroute-cache
!
interface FastEthernet1/0
  description TO SLB
  ip address 10.1.1.202 255.255.255.0
  ip directed-broadcast
  no ip mroute-cache
  duplex half
!

```

```

interface Virtual-Template1
  description GTP VIRTUAL TEMPLATE
  ip address 192.168.2.2 255.255.255.0
  encapsulation gtp
  gprs access-point-list gprs1
  !
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
!
gprs access-point-list gprs1
  access-point 1
    access-point-name gprs.company.com
    access-mode non-transparent
    ip-address-pool dhcp-proxy-client
    dhcp-server 10.100.0.5 10.100.0.6
    dhcp-gateway-address 10.27.3.1
  exit
!
gprs maximum-pdp-context-allowed 45000
gprs qos map canonical-qos
gprs gtp path-echo-interval 0
gprs dfp max-weight 32
gprs slb cef 10.10.10.10

```

GGSN3 Configuration Statements

```

service gprs ggsn
!
hostname GGSN3
!
ip dfp agent gprs
  port 1111
  password Cookies 0
  inservice
!
ip domain-name gprs.com
!
interface loopback 1
  description LOOPBACK SAME AS IOS SLB VSERVER ADDRESS
  ip address 10.10.10.10 255.255.255.255
  no ip route-cache
  no ip mroute-cache
!
interface FastEthernet1/0
  description TO SLB
  ip address 10.1.1.203 255.255.255.0
  ip directed-broadcast
  no ip mroute-cache
  duplex half
!
interface Virtual-Template1
  description GTP VIRTUAL TEMPLATE
  ip address 192.168.3.3 255.255.255.0
  encapsulation gtp
  gprs access-point-list gprs1
  !
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
!

gprs access-point-list gprs1
  access-point 1
    access-point-name gprs.company.com
    access-mode non-transparent

```

```

ip-address-pool dhcp-proxy-client
dhcp-server 10.100.0.5 10.100.0.6
dhcp-gateway-address 10.27.3.1
exit
!
gprs maximum-pdp-context-allowed 45000
gprs qos map canonical-qos
gprs gtp path-echo-interval 0
gprs dfp max-weight 32
gprs slb cef 10.10.10.10

```

IOS SLB with GTP Load Balancing and NAT Example

The following example uses the same basic configuration as in the “[IOS SLB with GTP Load Balancing Configuration Example](#)” section on page 12-17, including the network shown in [Figure 2](#), but with the addition of NAT:

- [IOS SLB Configuration Statements, page 12-22](#)
- [GGSN1 Configuration Statements, page 12-23](#)
- [GGSN2 Configuration Statements, page 12-24](#)
- [GGSN3 Configuration Statements, page 12-24](#)

IOS SLB Configuration Statements

```

hostname GTP_SLB
!
ip domain-name gprs.com
!
ip slb serverfarm GPRS
nat server
real 192.168.1.1
weight 1
faildetect numconns 1 numclients 1
inservice
!
real 192.168.2.2
weight 1
faildetect numconns 1 numclients 1
inservice
!
real 192.168.3.3
weight 1
faildetect numconns 1 numclients 1
inservice
!
ip slb vserver FOR_GPRS
virtual 10.10.10.10 udp 3386 service gtp
serverfarm GPRS
inservice
!

ip slb dfp password Cookies 0
agent 10.1.1.201 1111 30 0 10
agent 10.1.1.202 1111 30 0 10
agent 10.1.1.203 1111 30 0 10
!
interface FastEthernet1/0
description TO SERVERFARM GPRS

```

```

ip address 10.1.1.100 255.255.255.0
no ip redirects
duplex half
!
interface FastEthernet3/0
description TO SGSN
ip address 10.2.1.100 255.255.255.0
no ip mroute-cache
duplex half
!
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
ip route 192.168.1.1 255.255.255.255 10.1.1.201
ip route 192.168.2.2 255.255.255.255 10.1.1.202
ip route 192.168.3.3 255.255.255.255 10.1.1.203

```

GGSN1 Configuration Statements

```

service gprs ggsn
!
hostname GGSN1
!
ip dfp agent gprs
port 1111
password Cookies 0
inservice
!
ip domain-name gprs.com
!
interface FastEthernet1/0
description TO SLB
ip address 10.1.1.201 255.255.255.0
ip directed-broadcast
no ip mroute-cache
duplex half
!
interface Virtual-Template1
description GTP VIRTUAL TEMPLATE
ip address 192.168.1.1 255.255.255.0
encapsulation gtp
gprs access-point-list gprs1
!
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
!
gprs access-point-list gprs1
access-point 1
access-point-name gprs.company.com
access-mode non-transparent
ip-address-pool dhcp-proxy-client
dhcp-server 10.100.0.5 10.100.0.6
dhcp-gateway-address 10.27.3.1
exit
!

gprs maximum-pdp-context-allowed 45000
gprs qos map canonical-qos
gprs gtp path-echo-interval 0
gprs dfp max-weight 32

```

GGSN2 Configuration Statements

```

service gprs ggsn
!
hostname GGSN2
!
ip dfp agent gprs
port 1111
password Cookies 0
inservice
!
ip domain-name gprs.com
!
interface FastEthernet1/0
description TO SLB
ip address 10.1.1.202 255.255.255.0
ip directed-broadcast
no ip mroute-cache
duplex half
interface Virtual-Template1
description GTP VIRTUAL TEMPLATE
ip address 192.168.2.2 255.255.255.0
encapsulation gtp
gprs access-point-list gprs1
!
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
!
gprs access-point-list gprs1
access-point 1
access-point-name gprs.company.com
access-mode non-transparent
ip-address-pool dhcp-proxy-client
dhcp-server 10.100.0.5 10.100.0.6
dhcp-gateway-address 10.27.3.1
exit
!
gprs maximum-pdp-context-allowed 45000
gprs qos map canonical-qos
gprs gtp path-echo-interval 0
gprs dfp max-weight 32

```

GGSN3 Configuration Statements

```

service gprs ggsn
!
hostname GGSN3
!
ip dfp agent gprs
port 1111
password Cookies 0
inservice
!
ip domain-name gprs.com
!
interface FastEthernet1/0
description TO SLB
ip address 10.1.1.203 255.255.255.0
ip directed-broadcast
no ip mroute-cache
duplex half
!

```



```
interface Virtual-Template1
  description GTP VIRTUAL TEMPLATE
  ip address 192.168.3.3 255.255.255.0
  encapsulation gtp
  gprs access-point-list gprs1
  !
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
!
gprs access-point-list gprs1
  access-point 1
    access-point-name gprs.company.com
    access-mode non-transparent
    ip-address-pool dhcp-proxy-client
    dhcp-server 10.100.0.5 10.100.0.6
    dhcp-gateway-address 10.27.3.1
  exit
!
gprs maximum-pdp-context-allowed 45000
gprs qos map canonical-qos
gprs gtp path-echo-interval 0
gprs dfp max-weight 32
```

IOS SLB with GTP Load Balancing, NAT, and GTP Cause Code Inspection Example

The following example uses the same basic configuration as in the [“IOS SLB with GTP Load Balancing and NAT Example” section on page 12-22](#), including the network shown in [Figure 2](#), but with the GTP cause code inspection enabled. In this configuration:

- The GSN idle timer is set to 20 seconds.
- The GTP request idle timer is set to 15 seconds.
- The virtual server accepts PDP context creates only from International Mobile Subscriber IDs (IMSI) with carrier code **mcc 222 mnc 22**.

Following are the configuration statements for the configuration shown in [Figure 2](#), with the addition of NAT and GTP cause code inspection support:

- [IOS SLB Configuration Statements, page 12-26](#)
- [GGSN1 Configuration Statements, page 12-23](#) (no change for GTP cause code inspection)
- [GGSN2 Configuration Statements, page 12-24](#) (no change for GTP cause code inspection)
- [GGSN3 Configuration Statements, page 12-24](#) (no change for GTP cause code inspection)

IOS SLB Configuration Statements

```

hostname GTP_SLB
!
ip domain-name gprs.com
!
ip slb timers gtp gsn 20
!
ip slb serverfarm GPRS
  nat server
  real 192.168.1.1
    weight 1
    faildetect numconns 1 numclients 1
    inservice
  !
  real 192.168.2.2
    weight 1
    faildetect numconns 1 numclients 1
    inservice
  !
  real 192.168.3.3
    weight 1
    faildetect numconns 1 numclients 1
    inservice
  !
ip slb vserver FOR_GPRS
  virtual 10.10.10.10 udp 0 service gtp-inspect
  idle gtp request 15
  client gtp carrier-code mcc 222 mnc 22
  serverfarm GPRS
  inservice
!
ip slb dfp password Cookies 0
  agent 10.1.1.201 1111 30 0 10
  agent 10.1.1.202 1111 30 0 10
  agent 10.1.1.203 1111 30 0 10
!
interface FastEthernet1/0
  description TO SERVERFARM GPRS
  ip address 10.1.1.100 255.255.255.0
  no ip redirects
  duplex half
!
interface FastEthernet3/0
  description TO SGSN
  ip address 10.2.1.100 255.255.255.0
  no ip mroute-cache
  duplex half
!
ip route 10.111.111.111 255.255.255.255 FastEthernet1/0
ip route 192.168.1.1 255.255.255.255 10.1.1.201
ip route 192.168.2.2 255.255.255.255 10.1.1.202
ip route 192.168.3.3 255.255.255.255 10.1.1.203

```