Server Load Balancing Configuration Guide Cisco IOS Release 12.2SX

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
    800 553-NETS (6387)
Fax: 408 527-0883

C O N T E N T S

Information About Cisco IOS SLB

To configure IOS SLB, you should understand the following concepts:

Note Some IOS SLB features are specific to one platform and are not described in this feature document. For information about those features, refer to the appropriate platform-specific documentation.

Overview

This document describes how to configure the Cisco IOS Server Load Balancing (IOS SLB) feature. For a complete description of the IOS SLB commands in this chapter, refer to the "Server Load Balancing Commands" chapter of the *Cisco IOS IP Application Services Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

The SLB feature is a Cisco IOS-based solution that provides IP server load balancing. Using the IOS SLB feature:

1. The network administrator defines 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 are configured to connect to the IP address of the virtual server.
2. The virtual server IP address is configured as a loopback address, or secondary IP address, on each of the real servers.
3. When a client initiates a connection to the virtual server, the IOS SLB function chooses a real server for the connection based on a configured load-balancing algorithm.

The IOS SLB feature provides load balancing for a variety of networked devices and services, including:

- Application servers, such as Hypertext Transfer Protocol (HTTP), Telnet, File Transfer Protocol (FTP), and so on
- Firewalls
- Service nodes, such as authentication, authorization, and accounting (AAA) servers, web caches, and so on

In addition, the IOS SLB Exchange Director enables advanced load-balancing routing capabilities for the following additional service nodes:

- mobile Service Exchange Framework (mSEF) components:

    ◦ Cisco Content Services Gateways (CSGs)

If you are running with Supervisor Engine 32 (SUP32-MSFC2A), CSG Release 3.1(3)C7(1) or later is required.

- ◦ Cisco gateway general packet radio service (GPRS) support nodes (GGSNs)
    ◦ Cisco Service Selection Gateways (SSGs)
    ◦ Cisco Home Agents
- Other components for mobile, Public Wireless LAN (PWLAN), and Service Provider networks:

    ◦ Wireless Application Protocol (WAP) gateways
    ◦ Protocol optimization gateways
    ◦ Non-Cisco GGSNs and Home Agents
    ◦ Other RADIUS-aware flow gateways. These gateways are proxies or routing nodes that receive RADIUS Authorization and Accounting requests for users that route flows through the gateways. The Exchange Director binds the RADIUS and data flows to the same gateway, ensuring that the gateway receives a complete and consistent view of the network activity for the user.

The Exchange Director also adds the following features:

- Enhanced failover capabilities for single-chassis failover within the mSEF for Cisco Catalyst 6500 series switches and Cisco 7600 series routers. When used with Route Processor Redundancy Plus (RPR+), IOS SLB stateful backup for redundant route processors provides full IOS SLB stateful failover for these platforms.
- Flow persistence, which provides intelligent return routing of load-balanced IP flows.

The figure below illustrates a simple IOS SLB network.

Figure 1: Logical View of Cisco IOS SLB
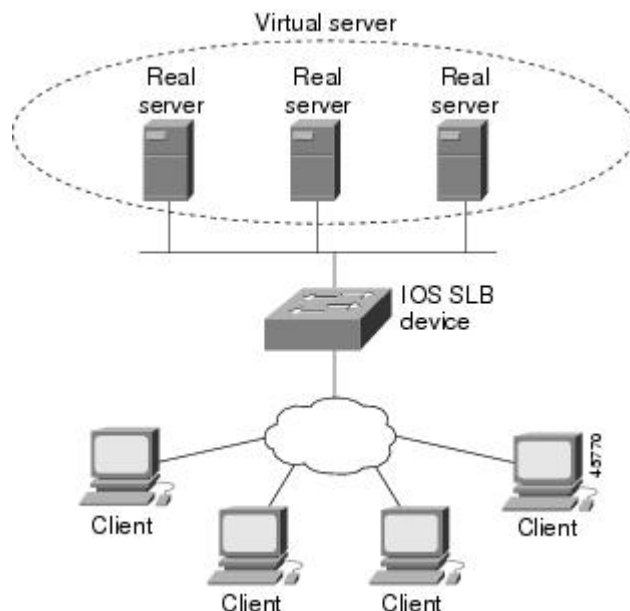
Benefits of IOS SLB

IOS SLB shares the same software code base as Cisco IOS and has all of the software features sets of Cisco IOS software.

On Cisco Catalyst 6500 series switches, IOS SLB uses hardware acceleration to forward packets at a very high speed when running in dispatched mode.

IOS SLB assures continuous, high availability of content and applications with techniques for actively managing servers and connections in a distributed environment. By distributing user requests across a cluster of servers, IOS SLB optimizes responsiveness and system capacity, and reduces the cost of providing Internet, database, and application services for large, medium, and small-scale sites.

IOS SLB facilitates scalability, availability, and ease of maintenance as follows:

- The addition of new physical (real) servers, and the removal or failure of existing servers, can occur at any time, transparently, without affecting the availability of the virtual server.
- IOS SLB's slow start capability allows a new server to increase its load gradually, preventing failures caused by assigning many new connections to the server in a short period.
- IOS SLB supports fragmented packets and packets with IP options, buffering your servers from client or network vagaries that are beyond your control.
- IOS SLB firewall load balancing enables you to scale access to your Internet site. You can add firewalls without affecting existing connections, enabling your site to grow without impacting customers.

Using DFP enables IOS SLB to provide weights to another load-balancing system. IOS SLB can act as a DFP manager, receiving weights from host servers, and it can act as a DFP agent, sending weights to a DFP manager. The functions are enabled independently--you can implement either one, or both, at the same time.

IOS SLB makes the administration of server applications easy. Clients know only about virtual servers; no administration is required for real server changes.

IOS SLB provides security for the real server because it never announces the real server's address to the external network. Users are familiar only with the virtual IP address. You can filter unwanted flows based on both IP address and TCP or UDP port numbers. Additionally, though it does not eliminate the need for a firewall, IOS SLB can help protect against some denial-of-service attacks.

In a branch office, IOS SLB allows balancing of multiple sites and disaster recovery in the event of full-site failure, and distributes the work of load balancing.

Cisco IOS SLB Features

Routing Features

Algorithms for Server Load Balancing

IOS SLB provides the following load-balancing algorithms:

You can specify one of these algorithms as the basis for choosing a real server for each new connection request that arrives at the virtual server.

For each algorithm, connections in the closing state are counted against the number of connections assigned to a real server. This impacts the least connections algorithm more than the other algorithms, because the least connections algorithm is influenced by the number of connections. IOS SLB adjusts the number of connections per real server, and the algorithm metrics, each time a connection is assigned.

Weighted Round Robin Algorithm

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 manage 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 To configure the IOS SLB device to use a round robin algorithm, assign a weight of n=1 to all of the servers in the server farm. GPRS load balancing *without* GTP cause code inspection enabled requires the weighted round robin algorithm. You can bind a server farm that uses weighted least connections to a virtual server providing GPRS load balancing without GTP cause code inspection enabled, but you cannot place the virtual server in service. If you try to do so, IOS SLB issues an error message. The Home Agent Director requires the weighted round robin algorithm. You can bind a server farm that uses weighted least connections to a Home Agent Director virtual server, but you cannot place the virtual server INSERVICE. If you try to do so, IOS SLB issues an error message. RADIUS load balancing requires the weighted round robin algorithm. RADIUS load balancing accelerated data plane forwarding *does not* support the weighted round robin algorithm.

Weighted Least Connections Algorithm

The weighted least connections algorithm specifies that the next real server chosen from a server farm 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 $n1/(n1+n2+n3...)$.

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. GPRS load balancing *without* GTP cause code inspection enabled *does not* support the weighted least connections algorithm. GPRS load balancing *with* GTP cause code inspection enabled *does* support the weighted least connections algorithm. Access Service Network (ASN) load balancing (for Mobile Station Pre-Attachment requests), the Home Agent Director, RADIUS load balancing, and RADIUS load balancing accelerated data plane forwarding *do not* support the weighted least connections algorithm.

Route Map Algorithm

The route map algorithm is valid only with IOS SLB RADIUS load balancing accelerated data plane forwarding, also known as Turbo RADIUS load balancing. Turbo RADIUS load balancing is a high-performance solution that uses policy-based routing (PBR) route maps to manage subscriber data-plane traffic in a Cisco Content Services Gateway (CSG) environment. When Turbo RADIUS load balancing receives a RADIUS payload, it inspects the payload, extracts the framed-IP attribute, applies a route map to the IP address, and then determines which CSG is to manage the subscriber.

For more information about policy-based routing, see the "Policy-Based Routing" and "Configuring Policy-Based Routing" sections of the *Cisco IOS IP Routing Configuration Guide* .

Note   RADIUS load balancing accelerated data plane forwarding requires the route map algorithm.

Bind ID Support

A bind ID allows one physical server to be bound to multiple virtual servers and report a different weight for each one. Thus, the single real server is represented as multiple instances of itself, each having a different bind ID. Dynamic Feedback Protocol (DFP) uses the bind ID to identify the instance of the real server for which a given weight is specified. Use the bind ID feature only if you are using DFP.

GPRS load balancing and the Home Agent Director do not support bind IDs.

Client-Assigned Load Balancing

Client-assigned load balancing allows you to limit access to a virtual server by specifying the list of client IP subnets that are permitted to use that virtual server. With this feature, you can assign a set of client IP subnets (such as internal subnets) connecting to a virtual IP address to one server farm or firewall farm, and assign another set of clients (such as external clients) to a different server farm or firewall farm.

GPRS load balancing and the Home Agent Director do not support client-assigned load balancing.

Connection Rate Limiting

IOS SLB enables you to specify the maximum connection rate allowed for a real server in a server farm. For more information, see the description of the **rate** command in real server configuration mode.

Content Flow Monitor Support

IOS SLB supports the Cisco Content Flow Monitor (CFM), a web-based status monitoring application within the CiscoWorks2000 product family. You can use CFM to manage Cisco server load-balancing devices. CFM runs on Windows NT and Solaris workstations, and is accessed using a web browser.

Delayed Removal of TCP Connection Context

Because of IP packet ordering anomalies, IOS SLB might "see" the end of a TCP connection (a finish [FIN] or reset [RST]) followed by other packets for the connection. This problem usually occurs if multiple paths exist for the TCP connection packets to follow. To correctly redirect the packets that arrive after the connection has ended, IOS SLB retains the TCP connection information, or context, for a specified length of time. The length of time the context is retained after the connection ends is controlled by a configurable delay timer.

Firewall Load Balancing

As its name implies, firewall load balancing:

- Enables IOS SLB to balance flows to firewalls.
- Uses a load-balancing device on each side of a group of firewalls (called a firewall farm) to ensure that the traffic for each flow travels to the same firewall, ensuring that the security policy is not compromised.

You can configure more than one firewall farm in each load-balancing device.

- Layer 3 firewalls--Have IP-addressable interfaces. Are supported by IOS SLB firewall load balancing if they are subnet-adjacent to the firewall load-balancing device and have unique MAC addresses. The device does not modify the IP addresses in the user packet. To send the packet to the chosen firewall, the device determines which interface to use and changes the Layer 2 headers accordingly. This type of routing is the standard dispatched routing used by IOS SLB.
- Layer 2 firewalls--Do not have IP addresses. Are transparent to IOS SLB firewall load balancing. IOS SLB supports Layer 2 firewalls by placing them between two IP-addressable interfaces.

Whereas many Layer 3 firewalls might exist off one Layer 3 interface on the load-balancing device (for example, one LAN), only one Layer 2 firewall can exist off each interface.

When configuring the load-balancing device, you configure a Layer 3 firewall using its IP address, and a Layer 2 firewall using the IP address of the interface of the device on the "other side" of the firewall.

To balance flows across the firewalls in a firewall farm, IOS SLB firewall load balancing performs a route lookup on each incoming flow, examining the source and destination IP addresses (and optionally the source and destination TCP or User Datagram Protocol [UDP] port numbers). Firewall load balancing applies a hash algorithm to the results of the route lookup and selects the best firewall to manage the connection request.

Note   IOS SLB firewall load balancing *must* examine incoming packets and perform route lookup. With Cisco Catalyst 6500 series switches, some additional packets might need to be examined. Firewall load balancing has an impact on internal (secure) side routing performance and must be considered in the complete design.

To maximize availability and resilience in a network with multiple firewalls, configure a separate equal-weight route to each firewall, rather than one route to only one of the firewalls.

IOS SLB firewall load balancing provides the following capabilities:

- Connections initiated from either side of the firewall farm are load-balanced.
- The load is balanced among a set of firewalls--the firewall farm.
- All packets for a connection travel through the same firewall. Subsequent connections can be "sticky," ensuring that they are assigned to the same firewall.
- Source-IP, destination-IP, and source-destination-IP sticky connections are supported.
- Probes are used to detect and recover from firewall failures.
- Redundancy is provided. Hot Standby Router Protocol (HSRP), stateless backup, and stateful backup are all supported.
- Multiple interface types and routing protocols are supported, enabling the external (Internet side) load-balancing device to act as an access router.
- Proxy firewalls are supported.

GTP IMSI Sticky Database

IOS SLB can select a GGSN for a given International Mobile Subscriber ID (IMSI), and forward all subsequent Packet Data Protocol (PDP) create requests from the same IMSI to the selected GGSN.

To enable this feature, IOS SLB uses a GTP IMSI sticky database, which maps each IMSI to its corresponding real server, in addition to its session database.

1 IOS SLB creates a sticky database object when it processes the first GTP PDP create request for a given IMSI.
2 IOS SLB removes the sticky object when it receives a notification to do so from the real server, or as a result of inactivity.

3 When the last PDP belonging to an IMSI is deleted on the GGSN, the GGSN notifies IOS SLB to remove the sticky object.

Home Agent Director

Home agents are the anchoring points for mobile nodes. They route flows for a mobile node to its current foreign agent (point of attachment).

The Home Agent Director load balances Mobile IP Registration Requests (RRQs) among a set of home agents (configured as real servers in a server farm). The Home Agent Director 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 home agents must be Layer 2-adjacent to the IOS SLB device.
- Does not support stateful backup. See the "Stateful Backup" section for more information.
- Delivers RRQs destined to the virtual Home Agent Director IP address to one of the real home agents, using the weighted round robin load-balancing algorithm. See the "Weighted Round Robin Algorithm" section for more information about this algorithm.
- Requires DFP in order to allocate RRQs based on capacity.

For more information about Mobile IP, home agents, and related topics, refer to the *Cisco IOS IP Mobility Configuration Guide*.

Interface Awareness

Some environments require IOS SLB to take into account the input interface when mapping packets to virtual servers, firewall farms, connections, and sessions. In IOS SLB, this function is called interface awareness. When interface awareness is configured, IOS SLB processes only traffic arriving on configured access interfaces. (An access interface is any Layer 3 interface.)

Such "sandwich" environments require IOS SLB on both sides of a farm of CSGs, SSGs, or firewalls. For example, you might want IOS SLB to perform RADIUS load balancing on one side of a farm and firewall load balancing on the other, or firewall load balancing on both sides of a firewall farm.

Maximum Connections

IOS SLB allows you to configure maximum connections for server and firewall load balancing.

- For server load balancing, you can configure a limit on the number of active connections that a real server is assigned. If the maximum number of connections is reached for a real server, IOS SLB automatically switches all further connection requests to other servers until the connection number drops below the specified limit.
- For firewall load balancing, you can configure a limit on the number of active TCP or UDP connections that a firewall farm is assigned. If the maximum number of connections is reached for the firewall farm, new connections are dropped until the connection number drops below the specified limit.

Multiple Firewall Farm Support

You can configure more than one firewall farm in each load-balancing device.

Network Address Translation

Cisco IOS Network Address Translation (NAT), RFC 1631, allows unregistered "private" IP addresses to connect to the Internet by translating them into globally registered IP addresses. As part of this functionality, Cisco IOS NAT can be configured to advertise only one address for the entire network to the outside world. This configuration provides additional security and network privacy, effectively hiding the entire internal network from the world behind that address. NAT has the dual functionality of security and address conservation, and is typically implemented in remote access environments.

Session Redirection

Session redirection NAT involves redirecting packets to real servers. IOS SLB can operate in one of two session redirection modes, dispatched mode or directed mode.

Note In both dispatched and directed modes, IOS SLB must track connections. Therefore, you must design your network so that there is no alternate network path from the real servers to the client that bypasses the load-balancing device.

Dispatched Mode

In dispatched NAT mode, the virtual server address is known to the real servers; you must configure the virtual server IP address as a loopback address, or secondary IP address, on each of the real servers. IOS SLB redirects packets to the real servers at the media access control (MAC) layer. Because the virtual server IP address is not modified in dispatched mode, the real servers must be Layer 2-adjacent to IOS SLB, or intervening routers might not be able to route to the chosen real server.

Refer to the "Configuring Virtual Interfaces" chapter of the *Cisco IOS Interface Configuration Guide* for more information about configuring the loopback address.

Note Some UDP applications cannot respond from the loopback interface. If that situation occurs, you must use directed mode.

Directed Mode

In directed NAT mode, the virtual server can be assigned an IP address that is not known to any of the real servers. IOS SLB translates packets exchanged between a client and a real server, using NAT to translate the virtual server IP address to a real server IP address.

IOS SLB supports the following types of NAT:

> ✎
>
> **Note** You can use both server NAT and client NAT for the same connection. IOS SLB does not support FTP or firewall load balancing in directed mode. Therefore, FTP and firewall load balancing cannot use NAT. IOS SLB supports only client NAT for TCP and UDP virtual servers. IOS SLB supports only server NAT (but not server port translation) for Encapsulation Security Payload (ESP) virtual servers or Generic Routing Encapsulation (GRE) virtual servers.

Server NAT

Server NAT involves replacing the virtual server IP address with the real server IP address (and vice versa). Server NAT provides the following benefits:

- Servers can be many hops away from the load-balancing device.
- Intervening routers can route to them without requiring tunnelling.
- Loopback and secondary interfaces are not required on the real server.
- The real server need not be Layer 2-adjacent to IOS SLB.
- The real server can initiate a connection to a virtual server on the same IOS SLB device.

Client NAT

If you use more than one load-balancing device in your network, replacing the client IP address with an IP address associated with one of the devices results in proper routing of outbound flows to the correct device. Client NAT also requires that the ephemeral client port be modified since many clients can use the same ephemeral port. Even in cases where multiple load-balancing devices are not used, client NAT can be useful to ensure that packets from load-balanced connections are not routed around the device.

Static NAT

With static NAT, address translations exist in the NAT translation table as soon as you configure static NAT commands, and they remain in the translation table until you delete the static NAT commands.

You can use static NAT to allow some users to use NAT and allow other users on the same Ethernet interface to continue with their own IP addresses. This option enables you to provide a default NAT behavior for real servers, differentiating between responses from a real server, and connection requests initiated by the real server.

For example, you can use server NAT to redirect Domain Name System (DNS) inbound request packets and outbound response packets for a real server, and static NAT to process connection requests from that real server.

> ✎
>
> **Note** Static NAT is not required for DNS, but we recommend it, because static NAT hides your real server IP addresses from the outside world.

IOS SLB supports the following static NAT options, configured using the **ip slb static** command:

- Static NAT with dropped connections--The real server is configured to have its packets dropped by IOS SLB, if the packets do not correspond to existing connections. This option is usually used in conjunction with the subnet mask or port number option on the **real** command in static NAT configuration mode, such that IOS SLB builds connections to the specified subnet or port, and drops all other connections from the real server.

- Static NAT with a specified address--The real server is configured to use a user-specified virtual IP address when translating addresses.
- Static NAT with per-packet server load balancing--The real server is configured such that IOS SLB is not to maintain connection state for packets originating from the real server. That is, IOS SLB is to use server NAT to redirect packets originating from the real server. Per-packet server load balancing is especially useful for DNS load balancing. IOS SLB uses DNS probes to detect failures in the per-packet server load-balancing environment.

> **Note** Static NAT with per-packet server load balancing does not load-balance fragmented packets.

- Static NAT with sticky connections--The real server is configured such that IOS SLB is not to maintain connection state for packets originating from the real server, unless those packets match a sticky object:

  - If IOS SLB finds a matching sticky object, it builds the connection.
  - If IOS SLB does not find a matching sticky object, it forwards the packets without building the connection.

IOS SLB uses the following logic when handling a packet from a real server:

SUMMARY STEPS

1. Does the packet match a real server?
2. Does the packet match an existing connection?
3. Is the real server configured to use static NAT?
4. Is the real server configured to have its packets dropped by IOS SLB, if the packets do not correspond to existing connections?
5. Is the real server configured for per-packet server load balancing?
6. Is the real server configured to maintain connection state for sticky connections?
7. Can IOS SLB find a matching sticky object?

DETAILED STEPS

Step 1 Does the packet match a real server?

- If no, IOS SLB has no interest in the packet.
- If yes, continue.

Step 2 Does the packet match an existing connection?

- If yes, IOS SLB uses NAT to redirect the packet, in accordance with the connection control block.
- If no, continue.

Step 3 Is the real server configured to use static NAT?

- If no, IOS SLB manages the packet as usual. This functionality is also called static NAT pass-through.
- If yes, continue.

Step 4 Is the real server configured to have its packets dropped by IOS SLB, if the packets do not correspond to existing connections?

- If yes, IOS SLB drops the packet.

- If no, continue.

Step 5   Is the real server configured for per-packet server load balancing?

- If yes, IOS SLB uses NAT to redirect the packet.
- If no, continue.

Step 6   Is the real server configured to maintain connection state for sticky connections?

- If no, IOS SLB builds the connection.
- If yes, IOS SLB searches for a matching sticky object. Continue.

Step 7   Can IOS SLB find a matching sticky object?

- If no, IOS SLB drops the packet.
- If yes, IOS SLB builds the connection.

Server Port Translation

Server port translation, also known as port address translation, or PAT, is a form of server NAT that involves the translation of virtual server ports instead of virtual server IP addresses. Virtual server port translation does not require translation of the virtual server IP address, but you can use the two types of translation together.

IOS SLB supports server port translation for TCP and UDP only.

Port-Bound Servers

Port-bound servers allow one virtual server IP address to represent one set of real servers for one service, such as HTTP, and a different set of real servers for another service, such as Telnet. When you define a virtual server, you must specify the TCP or UDP port managed by that virtual server. However, if you configure NAT on the server farm, you can also configure port-bound servers.

Packets destined for a virtual server address for a port that is not specified in the virtual server definition are not redirected.

IOS SLB supports both port-bound and non-port-bound servers, but port-bound servers are recommended.

IOS SLB firewall load balancing does not support port-bound servers.

Route Health Injection

By default, a virtual server's IP address is advertised (added to the routing table) when you bring the virtual server into service (using the **inservice** command). If there is a preferred host route to a website's virtual IP address, you can advertise that host route, but there is no guarantee that the IP address is available. However, you can use the **advertise** command to configure IOS SLB to advertise the host route only when IOS SLB has verified that the IP address is available. IOS SLB withdraws the advertisement when the IP address is no longer available. This function is known as route health injection.

Sticky Connections

You can use the optional **sticky** command to enable IOS SLB to force connections from the same client to the same load-balanced server within a server farm.

Sometimes, a client transaction can require multiple consecutive connections, which means new connections from the same client IP address or subnet must be assigned to the same real server. These connections are especially important in firewall load balancing, because the firewall might need to profile the multiple connections in order to detect certain attacks.

- IOS SLB supports source-IP sticky connections.
- Firewall load balancing supports source-IP, destination-IP, and source-destination-IP sticky connections.
- RADIUS load balancing supports calling-station-IP, framed-IP, and username sticky connections.

For firewall load balancing, the connections between the same client-server pair are assigned to the same firewall. New connections are considered to be sticky as long as the following conditions are met:

- The real server is in either OPERATIONAL or MAXCONNS_THROTTLED state.
- The sticky timer is defined on a virtual server or on a firewall farm.

This binding of new connections to the same server or firewall is continued for a user-defined period after the last sticky connection ends.

To get the client-server address sticky behavior needed for "sandwich" firewall load balancing, you must enable sticky on both sides of the firewall farm. In this configuration, client-server sticky associations are created when an initial connection is opened between a client-server address pair. After this initial connection is established, IOS SLB maintains the sticky association in the firewall load-balancing devices on either side of the farm, and applies the sticky association to connections initiated from either the client or server IP address, by both firewall load-balancing devices.

Client subnet sticky is enabled when you specify the **sticky** command with a subnet mask. Subnet sticky is useful when the client IP address might change from one connection to the next. For example, before reaching IOS SLB, the client connections might pass through a set of NAT or proxy firewalls that have no sticky management of their own. Such a situation can result in failed client transactions if the servers do not have the logic to cope with it. In cases where such firewalls assign addresses from the same set of subnets, IOS SLB's sticky subnet mask can overcome the problems that they might cause.

Sticky connections also permit the coupling of services that are managed by more than one virtual server or firewall farm. This option allows connection requests for related services to use the same real server. For example, web server (HTTP) typically uses TCP port 80, and HTTPS uses port 443. If HTTP virtual servers and HTTPS virtual servers are coupled, connections for ports 80 and 443 from the same client IP address or subnet are assigned to the same real server.

Virtual servers that are in the same sticky group are sometimes called buddied virtual servers.

The Home Agent Director does not support sticky connections.

TCP Session Reassignment

IOS SLB tracks each TCP SYNchronize sequence number, or SYN, sent to a real server by a client attempting to open a new connection. If several consecutive SYNs are not answered, or if a SYN is replied to with an RST, the TCP session is reassigned to a new real server. The number of SYN attempts is controlled by a configurable reassign threshold.

IOS SLB firewall load balancing does not support TCP session reassignment.

Transparent Web Cache Load Balancing

IOS SLB can load-balance HTTP flows across a cluster of transparent web caches. To set up this function, configure the subnet IP addresses served by the transparent web caches, or some common subset of them,

as virtual servers. Virtual servers used for transparent web cache load balancing do not answer pings on behalf of the subnet IP addresses, and they do not affect traceroute.

In some cases, such as when its cache does not contain needed pages, a web cache must initiate its own connections to the Internet. Those connections should not be load-balanced back to the same set of web caches. To address this need, IOS SLB allows you to configure **client exclude** statements, which exclude connections initiated by the web caches from the load-balancing scheme.

IOS SLB firewall load balancing does not support transparent web cache load balancing.

Security Features

Alternate IP Addresses

IOS SLB enables you to telnet to the load-balancing device using an alternate IP address. To do so, use either of the following methods:

- Use any of the interface IP addresses to telnet to the load-balancing device.
- Define a secondary IP address to telnet to the load-balancing device.

This function is similar to that provided by the LocalDirector (LD) Alias command.

Avoiding Attacks on Server Farms and Firewall Farms

IOS SLB relies on a site's firewalls to protect the site from attacks. In general, IOS SLB is no more susceptible to direct attack than is any switch or router. However, a highly secure site can take the following steps to enhance its security:

- Configure real servers on a private network to keep clients from connecting directly to them. This configuration ensures that the clients must go through IOS SLB to get to the real servers.
- Configure input access lists on the access router or on the IOS SLB device to deny flows from the outside network aimed directly at the interfaces on the IOS SLB device. That is, deny *all* direct flows from unexpected addresses.
- To protect against attackers trying to direct flows to real or nonexistent IP addresses in the firewall subnet, configure the firewalls in a private network.
- Configure firewalls to deny *all* unexpected flows targeted at the firewalls, especially flows originating from the external network.

Slow Start

To prevent an overload, slow start controls the number of new connections that are directed to a real server that has just been placed in service. In an environment that uses weighted least connections load balancing, a real server that is placed in service initially has no connections, and could therefore be assigned so many new connections that it becomes overloaded.

GPRS load balancing and the Home Agent Director do not support slow start.

SynGuard

SynGuard limits the rate of TCP start-of-connection packets (SYNchronize sequence numbers, or SYNs) managed by a virtual server to prevent a type of network problem known as a SYN flood denial-of-service attack. A user might send a large number of SYNs to a server, which could overwhelm or crash the server, denying service to other users. SynGuard prevents such an attack from bringing down IOS SLB or a real server. SynGuard monitors the number of SYNs managed by a virtual server at specific intervals and does not allow the number to exceed a configured SYN threshold. If the threshold is reached, any new SYNs are dropped.

IOS SLB firewall load balancing and the Home Agent Director do not support SynGuard.

Server Failure Detection and Recovery Features

Automatic Server Failure Detection

IOS SLB automatically detects each failed Transmission Control Protocol (TCP) connection attempt to a real server, and increments a failure counter for that server. (The failure counter is not incremented if a failed TCP connection from the same client has already been counted.) If a server's failure counter exceeds a configurable failure threshold, the server is considered out of service and is removed from the list of active real servers.

For RADIUS load balancing, the IOS SLB performs automatic server failure detection when a RADIUS request is not answered by the real server.

If you have configured all-port virtual servers (that is, virtual servers that accept flows destined for all ports except GTP ports), flows can be passed to servers for which no application port exists. When the servers reject these flows, IOS SLB might fail the servers and remove them from load balancing. This situation can also occur in slow-to-respond AAA servers in RADIUS load-balancing environments. To prevent this situation, you can disable automatic server failure detection.

Note   If you disable automatic server failure detection using the **no faildetect inband** command, we strongly recommend that you configure one or more probes. If you specify the **no faildetect inband** command, the **faildetect numconns** command is ignored, if specified.

Automatic Unfail

When a real server fails and is removed from the list of active servers, it is assigned no new connections for a length of time specified by a configurable retry timer. After that timer expires, the server is again eligible

for new virtual server connections and IOS SLB sends the server the next qualifying connection. If the connection is successful, the failed server is placed back on the list of active real servers. If the connection is unsuccessful, the server remains out of service and the retry timer is reset. The unsuccessful connection must have experienced at least one retry, otherwise the next qualifying connection is also sent to that failed server.

Backup Server Farms

A backup server farm is a server farm that can be used when none of the real servers defined in a primary server farm is available to accept new connections. When configuring backup server farms, keep in mind the following considerations:

- A server farm can act as both primary and backup at the same time.
- The same real server cannot be defined in both primary and backup at the same time.
- Both primary and backup require the same NAT configuration (none, client, server, or both). In addition, if NAT is specified, both server farms must use the same NAT pool.

Dynamic Feedback Protocol (DFP) Agent Subsystem Support

IOS SLB supports the DFP Agent Subsystem feature, also called global load balancing, which enables client subsystems other than IOS SLB to act as DFP agents. With the DFP Agent Subsystem, you can use multiple DFP agents from different client subsystems at the same time.

For more information about the DFP Agent Subsystem, refer to the *DFP Agent Subsystem* feature document for Cisco IOS Release 12.2(18)SXD.

DFP for Cisco IOS SLB

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 a sudden change occurs in a real server's status.

The weights calculated by DFP override the static weights you define using the **weight** command in server farm configuration mode. 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, or as both at the same time. In such a configuration, IOS SLB sends periodic reports to the other DFP manager, 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) at the same time.

DFP and GPRS Load Balancing

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

approximation, DFP calculates the weight as the number of existing PDP contexts divided by the maximum allowed PDP contexts:

(existing PDP contexts)/(maximum PDP contexts)

Maximum PDP contexts are specified using the **gprs maximum-pdp-context-allowed** command, which defaults to 10,000 PDP contexts. If you accept the default value, DFP might calculate a very low weight for the GGSN:

(existing PDP contexts)/10000 = Low GGSN weight

When you specify maximum PDP contexts using the **gprs maximum-pdp-context-allowed** command, keep this calculation in mind. For example, Cisco 7200 series routers acting as GGSNs are often configured with a maximum of 45,000 PDP contexts.

DFP and the Home Agent Director

When using the Home Agent Director, you can define IOS SLB as a DFP manager and define a DFP agent on each home agent in the server farm, and the DFP agent can report the weights of the home agents. The DFP agents calculate the weight of each home agent based on CPU use, processor memory, and the maximum number of bindings that can be activated for each home agent:

(maximum-number-of-bindings - current-number-of-bindings)/maximum-number-of-bindings * (cpu-use + memory-use)/32 * maximum-DFP-weight = reported-weight

The *maximum-number-of-bindings* is 235,000. The *maximum-DFP-weight* is 24.

GGSN-IOS SLB Messaging

You can enable a GGSN to notify IOS SLB when certain conditions occur. The notifications enable IOS SLB to make intelligent decisions, which in turn improves GPRS load balancing and failure detection.

The notifications sent by the GGSN use GTP with message types from the unused space (reserved for future use) and the following information elements (IEs):

- Notification type, which indicates the notification condition. For example, this could be a notification to IOS SLB to reassign the session to an alternate GGSN, when the current GGSN fails on Call Admission Control (CAC).
- Identifier of the relevant session (session key).
- Other IEs specific to the notification type. For example, for a notification to reassign, GGSN includes the create response, which it would otherwise have sent to the SGSN. This enables IOS SLB to relay this response back to SGSN when the maximum number of reassignments due to notification reach the configured limit.

GGSN-IOS SLB messaging is supported in both dispatched mode and directed modes.

INOP_REAL State for Virtual Servers

You can configure a virtual server such that, if all of the real servers that are associated with the virtual server are inactive, the following actions occur:

- The virtual server is placed in the INOP_REAL state.
- An SNMP trap is generated for the virtual server's state transition.
- The virtual server stops answering ICMP requests.

For more information, see the description of the **inservice (server farm virtual server)** command in SLB server farm virtual server configuration mode.

Probes

Probes determine the status of each real server in a server farm, or each firewall in a firewall farm. The Cisco IOS SLB feature supports DNS, HTTP, ping, TCP, custom UDP, and WSP probes:

- A DNS probe sends domain name resolve requests to real servers, and verifies the returned IP addresses.
- An HTTP probe establishes HTTP connections to real servers, sends HTTP requests to the real servers, and verifies the responses. HTTP probes are a simple way to verify connectivity for devices that are server load-balanced, and for firewalls being firewall load-balanced (even devices on the other side of a firewall).

HTTP probes also enable you to monitor applications that are server load-balanced. With frequent probes, the operation of each application is verified, not just connectivity to the application.

HTTP probes do not support HTTP over Secure Socket Layer (HTTPS). That is, you cannot send an HTTP probe to an SSL server.

- A ping probe pings real servers. Like HTTP probes, ping probes are a simple way to verify connectivity for devices and firewalls being load-balanced.
- A TCP probe establishes and removes TCP connections. Use TCP probes to detect failures on TCP port 443 (HTTPS).
- A custom UDP probe can to support a variety of applications and protocols, including:
  - RADIUS Accounting/Authorization probes
  - GTP Echo probes
  - Connectionless WSP probes
  - XML-over-UDP probes for CSG user-database load-balancing
  - Mobile IP RRQ/RRP
- A WSP probe simulates requests for wireless content and verifies the retrieved content. Use WSP probes to detect failures in the Wireless Application Protocol (WAP) stack on port 9201.

You can configure more than one probe, in any combination of supported types, for each server farm, or for each firewall in a firewall farm.

You can also flag a probe as a routed probe, with the following considerations:

- Only one instance of a routed probe per server farm can run at any given time.
- Outbound packets for a routed probe are routed directly to a specified IP address.

IOS SLB probes use the SA Agent. You might want to specify the amount of memory that the SA Agent can use, using the **rtr low-memory** command. If the amount of available free memory falls below the value specified in the **rtr low-memory** command, then the SA Agent does not allow new operations to be configured. For more details, see the description of the **rtr low-memory** command in the *Cisco IOS IP SLAs Command Reference*.

Probes in Server Load Balancing

Probes determine the status of each real server in a server farm. All real servers associated with all virtual servers tied to that server farm are probed.

If a real server fails for one probe, it fails for all probes. After the real server recovers, all probes must acknowledge its recovery before it is restored to service.

Note   If a probe is configured for stateful backup and a failover occurs, the change in status (from backup to active) is reflected accurately in the probe in the new active IOS SLB device. However, the probe in the new backup IOS SLB device (which had been the active device before the failover) still shows its status as active.

Probes in Firewall Load Balancing

Probes detect firewall failures. All firewalls associated with the firewall farm are probed.

If a firewall fails for one probe, it is failed for all probes. After the firewall recovers, all probes must acknowledge its recovery before the probe is restored to service.

To prevent password problems, make sure you configure the HTTP probe to expect status code 401. For more details, see the description of the **expect** command.

Use the **ip http server** command to configure an HTTP server on the device. For more details, see the description of the **ip http server** command in the *Cisco IOS Configuration Fundamentals Command Reference*.

In a transparent web cache load-balancing environment, an HTTP probe uses the real IP address of the web cache, since there is no virtual IP address configured.

Protocol Support Features

Protocol Support

IOS SLB supports the following protocols:

- Access Service Network (ASN)
- Domain Name System (DNS)
- Encapsulation Security Payload (ESP)
- File Transfer Protocol (FTP)
- Generic Routing Encapsulation (GRE)
- GPRS Tunneling Protocol v0 (GTP v0)
- GPRS Tunneling Protocol v1 (GTP v1)
- GPRS Tunneling Protocol v2 (GTP v2)
- Hypertext Transfer Protocol (HTTP)
- Hypertext Transfer Protocol over Secure Socket Layer (HTTPS)
- Internet Message Access Protocol (IMAP)
- Internet Key Exchange (IKE, was ISAKMP)
- IP in IP Encapsulation (IPinIP)

- Mapping of Airline Traffic over IP, Type A (MATIP-A)
- Network News Transport Protocol (NNTP)
- Post Office Protocol, version 2 (POP2)
- Post Office Protocol, version 3 (POP3)
- RealAudio/RealVideo through RTSP
- Remote Authentication Dial-In User Service (RADIUS)
- Simple Mail Transport Protocol (SMTP)
- Telnet
- Transmission Control Protocol (TCP) and standard TCP protocols
- User Datagram Protocol (UDP) and standard UDP protocols
- X.25 over TCP (XOT)
- Wireless Application Protocol (WAP), including:

  ◦ Connectionless Secure WSP
  ◦ Connectionless WSP
  ◦ Connection-Oriented Secure WSP
  ◦ Connection-Oriented WSP

AAA Load Balancing

IOS SLB provides RADIUS load-balancing capabilities for RADIUS authentication, authorization, and accounting (AAA) servers.

IOS SLB provides the following RADIUS load-balancing functions:

- Balances RADIUS requests among available RADIUS servers and proxy servers.
- Routes RADIUS request retransmissions (such as retransmissions of unanswered requests) to the same RADIUS server or proxy server as the original request.
- Provides session-based automatic failure detection.
- Supports both stateless backup and stateful backup.

In addition, IOS SLB can load-balance devices that proxy the RADIUS Authorization and Accounting flows in both traditional and mobile wireless networks. For more information, see the "RADIUS Load Balancing" section.

Audio and Video Load Balancing

IOS SLB can balance RealAudio and RealVideo streams through Real-Time Streaming Protocol (RTSP), for servers running RealNetworks applications.

VPN Server Load Balancing

IOS SLB can balance Virtual Private Network (VPN) flows, including the following flows:

- IP Security (IPSec) flows. An IPSec flow consists of a UDP control session and an ESP tunnel.
- Point-to-Point Tunneling Protocol (PPTP) flows. A PPTP flow consists of a TCP control session and a GRE tunnel.

Redundancy Features

An IOS SLB device can represent one point of failure, and the servers can lose their connections to the backbone, if either of the following occurs:

- The IOS SLB device fails.
- A link from a switch to the distribution-layer switch becomes disconnected.

To reduce that risk, IOS SLB supports the following redundancy enhancements, based on HSRP:

Stateless Backup

Stateless backup provides high network availability by routing IP flows from hosts on Ethernet networks without relying on the availability of one Layer 3 switch. Stateless backup is particularly useful for hosts that do not support a router discovery protocol (such as the Intermediate System-to-Intermediate System [IS-IS] Interdomain Routing Protocol [IDRP]) and do not have the functionality to shift to a new Layer 3 switch when their selected Layer 3 switch reloads or loses power.

Stateful Backup

Stateful backup enables IOS SLB to incrementally backup its load-balancing decisions, or "keep state," between primary and backup switches. The backup switch keeps its virtual servers in a dormant state until HSRP detects failover; then the backup (now primary) switch begins advertising virtual addresses and processing flows. You can use HSRP to configure a timer for failure detection.

Stateful backup provides IOS SLB with a one-to-one stateful or idle backup scheme. This means that only one instance of IOS SLB is handling client or server flows at a given time, and that there is at most one backup platform for each active IOS SLB switch.

The Home Agent Director do not support stateful backup.

Note   If a probe is configured for stateful backup and a failover occurs, the change in status (from backup to active) is reflected accurately in the probe in the new active IOS SLB device. However, the probe in the new backup IOS SLB device (which had been the active device before the failover) still shows its status as active.

Active Standby

Active standby enables two IOS SLBs to load-balance the same virtual IP address while at the same time acting as backups for each other. If a site has only one virtual IP address to load-balance, an access router is used to direct a subset of the flows to each IOS SLB using policy-based routing.

IOS SLB firewall load balancing supports active standby. That is, you can configure two pairs of firewall load balancing devices (one pair on each side of the firewalls), with each device in each pair handling traffic and backing up its partner.

Exchange Director Features

IOS SLB supports the Exchange Director for the mobile Service Exchange Framework (mSEF) for CiscoCisco 7600 series routers. The Exchange Director provides the following features:

ASN Load Balancing

IOS SLB can provide load balancing across a set of Access Service Network (ASN) gateways. The gateway server farm appears to the base station as one ASN gateway.

When a Mobile Subscriber Station (MSS) wants to enter the network, the base station sends a Mobile Station Pre-Attachment request to the virtual IP address of the IOS SLB. IOS SLB selects an ASN gateway and forwards the request to that gateway. The gateway responds directly to the base station with a Mobile Station Pre-Attachment response. If configured to do so, the base station then returns a Mobile Station Pre-Attachment ACK to IOS SLB, which forwards the ACK to the selected gateway. Thereafter, all subsequent transactions flow between the base station and the gateway.

If sticky connections are enabled for the ASN gateways, IOS SLB makes a load-balancing decision once for a subscriber and then forwards all subsequent requests from the same subscriber to the same Cisco Broadband Wireless Gateway (BWG). The sticky information is replicated to the standby IOS SLB.

IOS SLB populates the sticky database with Mobile Stations IDs (MSIDs), with one sticky entry for each MSS. The sticky database enables IOS SLB to perform persistent session tracking of the real server selected for the MSID. The first packet sent to a virtual IP address from an MSS creates the session object and the sticky object. Subsequent packets from the MSS use the MSID to find the real server in the sticky database, if the session lookup fails. All packets that belong to a given MSS are load-balanced to same BWG as long as the sticky object exists.

Redundancy support has been provided by replicating the sticky MSID entries to the backup IOS SLB. Redundancy works in both the intra-chassis (stateful switchover) and inter-chassis (HSRP) environments. Sessions need not be replicated to the standby IOS SLB.

GPRS Load Balancing

GPRS is the packet network infrastructure based on the European Telecommunications Standards Institute (ETSI) Global System for Mobile Communication (GSM) phase 2+ standards for transferring packet data from the GSM mobile user to the packet data network (PDN). The Cisco gateway GPRS support node (GGSN) interfaces with the serving GPRS support node (SGSN) using the GTP, which in turn uses UDP for transport. IOS SLB provides GPRS load balancing and increased reliability and availability for the GGSN.

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 assigns all PDP context creates from a specific IMSI to the same GGSN.
- IOS SLB supports GTP Version 0 (GTP v0), Version 1 (GTP v1), and Version 2 (GTP v2). Support for GTP enables IOS SLB to become "GTP aware," extending IOS SLB knowledge into Layer 5.
- GPRS load balancing maps enable IOS SLB to categorize and route user traffic based on access point names (APNs).

IOS SLB supports two types of GPRS load balancing:

GPRS Load Balancing without GTP Cause Code Inspection

GPRS 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.
- Supports stateful backup only if sticky connections are enabled. See the "Stateful Backup" section for more information.
- 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 Algorithm" section for more information about this algorithm.
- Requires DFP in order to account for secondary PDP contexts in GTP v1 and GTP v2.

GPRS Load Balancing with GTP Cause Code Inspection

GPRS 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.

The table below 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

Cause Code	IOS SLB Action
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

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

- Must operate in directed server NAT mode.
- Supports stateful backup. See the "Stateful Backup" section for more information.
- Tracks the number of open PDP contexts for each GGSN, which enables GGSN server farms to use the weighted least connections (**leastconns**) algorithm for GPRS load balancing. See the "Weighted Least Connections Algorithm" section 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 account for secondary PDP contexts even without DFP.

Dual-Stack Support for GTP Load Balancing

IPv6 support enables IOS SLB to manage IPv6 addresses for GTP load balancing, for all versions of GTP (v0, v1, v2).

Dual-stack support enables IOS SLB to manage dual-stack implementations for GTP load balancing. A dual stack implementation is one that uses both IPv4 and IPv6 addresses.

When configuring dual-stack support for GTP load balancing, keep the following considerations in mind:

- The real server must be configured as a dual-stack real server, with the IPv4 and IPv6 addresses, using the **real** command in SLB server farm configuration mode.
- The virtual server must be configured as a dual-stack virtual server, with the virtual IPv4 and IPv6 addresses and the optional IPv6 prefix, using the **virtual** command in SLB virtual server configuration mode.

- To specify the primary IPv6 server farm and optional backup IPv6 server farm, use the **serverfarm** command in SLB virtual server configuration mode.
- The **client** command in SLB virtual server configuration mode is not supported.
- The gateway must be configured with the IPv4 and IPv6 addresses of the virtual server.
- The interface between IOS SLB and the gateway must be configured with dual-stack addresses.
- All HSRP instances (both IPv4 and IPv6) for the client-facing interface must have the same HSRP state.

Home Agent Director

The Home Agent Director load balances Mobile IP Registration Requests (RRQs) among a set of home agents (configured as real servers in a server farm). Home agents are the anchoring points for mobile nodes. Home agents route flows for a mobile node to its current foreign agent (point of attachment).

The Home Agent Director 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 home agents must be Layer 2-adjacent to the IOS SLB device.
- Does not support stateful backup. See the "Stateful Backup" section for more information.
- Delivers RRQs destined to the virtual Home Agent Director IP address to one of the real home agents, using the weighted round robin load-balancing algorithm. See the "Weighted Round Robin Algorithm" section for more information about this algorithm.
- Requires DFP in order to allocate RRQs based on capacity.

For more information about Mobile IP, home agents, and related topics, refer to the *Cisco IOS IP Configuration Guide*, Release 12.2.

KeepAlive Application Protocol (KAL-AP) Agent Support

Support for the KeepAlive Application Protocol (KAL-AP) agent support enables IOS SLB to perform load balancing in a global server load balancing (GSLB) environment. KAL-AP provides load information along with its keepalive response message to the KAL-AP manager or GSLB device, such as the Global Site Selector (GSS), and helps the GSLB device load-balance client requests to the least-loaded IOS SLB devices.

When configuring KAL-AP agent support for IOS SLB, keep the following considerations in mind:

- KAL-AP agent support automatically detects the Virtual Private Network (VPN) routing and forwarding (VRF) ID of an incoming request packet, and uses the same VRF ID when sending a response.
- A client that uses DNS caching might contact IOS SLB directly, instead of sending requests through the GSS. Therefore, configure the DNS setting in the client to avoid such a situation.

KAL-AP calculates the load value in one of two ways: relatively or absolutely. (IOS SLB CPU/memory load might affect the final KAL-AP load value.)

Relative KAL-AP Load Value

If the **farm-weight**command is not configured in server farm configuration mode, or if DFP is not enabled for the IOS SLB, KAL-AP calculates a relative load value, using the following formula:

KAL-AP Load = **256** - (number-*of-active-real-servers* * **256**/*number-of-inservice-real-servers*)

For example, if a site is provisioned with two real servers, and both real servers are inservice but only one is currently active, the resulting KAL-AP load value for that site is:

KAL-AP Load = **256** - (**1** * **256/2**) = **256** - **128** = **128**

Absolute KAL-AP Load Value

If the **farm-weight** command is configured in server farm configuration mode, and DFP is enabled for the IOS SLB, KAL-AP calculates an absolute load value, using the following formula:

KAL-AP Load = **256** - (*sum-of-max-dfp-weights-of-real-servers* * **256**/*farm-weight*)

Note   The maximum DFP weight for a real server is configured using the **gprs dfp max-weight** command in global configuration mode. However, the actual maximum DFP weight reported to KAL-AP is proportional to the load on the GGSN. For example, if a GGSN is configured with a maximum DFP weight of 100, but the GGSN is 50 percent loaded, it reports a maximum DFP weight of 50 to KAL-AP. If the DFP connection to the real server is down, KAL-AP uses the setting of the **weight** command in SLB real server configuration mode. If no **weight** command is configured for the real server, KAL-AP uses the default weight of 8.

For example, consider a site with the following settings:

- A server farm configured with a farm weight of 200.
- GGSN-1 configured with a maximum DFP weight of 100, 0 percent loaded (so it reports a DFP weight of 100).
- GGSN-2 configured with a maximum DFP weight of 100, 50 percent loaded (so it reports a DFP weight of only 50).

The resulting KAL-AP load value for that site is:

KAL-AP Load = **256** - [(**100** + **50**) * **256/200**] = **256** - **192** = **64**

For best results, configure a **farm-weight** that is equal to the sum of the maximum DFP weights for the real servers in the server farm. For example, if there are three real servers in a server farm, configured with maximum DFP weights of 100, 50, and 50, then configure a **farm-weight** of 200 (that is, 100 + 50 + 50). If a real server is added to or removed from the server farm, you must adjust the **farm-weight** accordingly.

RADIUS Load Balancing

IOS SLB provides RADIUS load-balancing capabilities for RADIUS servers. In addition, IOS SLB can load-balance devices that proxy the RADIUS Authorization and Accounting flows in both traditional and mobile wireless networks, if desired. IOS SLB does this by correlating data flows to the same proxy that processed the RADIUS for that subscriber flow.

IOS SLB provides RADIUS load balancing in mobile wireless networks that use service gateways, such as the Cisco Service Selection Gateway (SSG) or the Cisco Content Services Gateway (CSG). The following mobile wireless networks are supported:

- GPRS networks. In a GPRS mobile wireless network, the RADIUS client is typically a GGSN.
- Simple IP CDMA2000 networks. CDMA2000 is a third-generation (3-G) version of Code Division Multiple Access (CDMA). In a simple IP CDMA2000 mobile wireless network, the RADIUS client is a Packet Data Service Node (PDSN).

- Mobile IP CDMA2000 networks. In a Mobile IP CDMA2000 mobile wireless network, both the Home Agent (HA) and the PDSN/Foreign Agent (PDSN/FA) are RADIUS clients.

IOS SLB provides the following RADIUS load-balancing functions:

- Balances RADIUS requests among available RADIUS servers and proxy servers.
- Routes RADIUS request retransmissions (such as retransmissions of unanswered requests) to the same RADIUS server or proxy server as the original request.
- Routes all of a subscriber's RADIUS flows, as well as all non-RADIUS data flows for the same subscriber, to the same service gateway.
- Supports multiple service gateway server farms (for example, one farm of SSGs and another of CSGs). IOS SLB examines the input interface in a packet to route it to the correct service gateway server farm.
- Supports multiple WAP gateway server farms behind a RADIUS load balancing virtual server, using RADIUS calling station IDs and usernames to select specific server farms. This enhancement enables RADIUS load balancing on both the control plane and the data plane. RADIUS load balancing on the control plane enables RADIUS messages to be load-balanced to AAA servers for subscriber authorization, authentication and accounting. RADIUS load balancing on the data plane enables data flows for a subscriber to maintain a consistent network path to the destination network device. In addition, the RADIUS virtual server can acknowledge RADIUS accounting messages and build or delete sticky objects, rather than having to forward the messages to the specified server.
- Can route data packets to a real server in the CSG farm, then to a real server in the SSG farm.
- Routes RADIUS Accounting-Request messages from a RADIUS client to the service gateway that processed the RADIUS Access-Request message for the subscriber. The service gateway can then clean up the host entry it has created for the subscriber.
- Uses the weighted round robin load-balancing algorithm. See the "Weighted Round Robin Algorithm" section for more information about this algorithm.
- Facilitates SSG single sign-on through the RADIUS protocol.
- Provides session-based automatic failure detection.
- Supports both stateless backup and stateful backup.

To perform RADIUS load balancing, IOS SLB uses the following RADIUS sticky databases:

- The IOS SLB RADIUS framed-IP sticky database associates each subscriber's IP address with a specific service gateway. In a GPRS mobile wireless network, IOS SLB uses the RADIUS framed-IP sticky database to route packets correctly.

Note   Subscriber IP addresses are assigned by service gateways or by RADIUS clients. If subscriber IP addresses are assigned from disjoint per-service gateway pools (so that the next-hop service gateway can be chosen based on the source IP address), IOS SLB can use policy routing to route subscriber flows.

- The IOS SLB RADIUS calling-station-ID sticky database associates each subscriber's calling station ID with a specific service gateway.
- The IOS SLB RADIUS username sticky database associates each subscriber's username with a specific service gateway.
- RADIUS load balancing maps enable IOS SLB to categorize and route user traffic based on RADIUS calling station IDs and usernames. RADIUS load balancing maps is mutually exclusive with Turbo RADIUS load balancing and RADIUS load balancing accounting local acknowledgement.
- RADIUS load balancing accounting local acknowledgement:

  ◦ Enables IOS SLB to respond to RADIUS accounting packets with an ACK response while maintaining sticky objects for those sessions.

- ◦ Is mutually exclusive with RADIUS load balancing maps and Turbo RADIUS load balancing.
- • In a CDMA2000 mobile wireless network, to route packets correctly, IOS SLB requires both the RADIUS framed-IP sticky database and either the RADIUS username sticky database or the RADIUS calling-station-ID sticky database.
- • The IOS SLB RADIUS International Mobile Subscriber ID (IMSI) sticky database maps the IMSI address for each user to the corresponding gateway. This enables IOS SLB to forward all subsequent flows for the same user to the same gateway.

RADIUS Load Balancing Accelerated Data Plane Forwarding

RADIUS load balancing accelerated data plane forwarding, also known as Turbo RADIUS load balancing, is a high-performance solution that uses basic policy-based routing (PBR) route maps to manage subscriber data-plane traffic in a Cisco Content Services Gateway (CSG) environment.

When Turbo RADIUS load balancing receives a RADIUS payload, it takes the following actions:

1. Inspects the payload.
2. Extracts the framed-IP attribute.
3. Applies a route map to the IP address.
4. Determines which CSG is to manage the subscriber.

If vendor-specific attribute (VSA) correlation is configured, and if the Cisco VSA is buffered, then the Cisco VSA is injected into the RADIUS Accounting-Start packet.

Turbo RADIUS load balancing does not require VSA correlation, but it does require a server farm configured with **predictor route-map** on the accounting virtual server.

> ✎
>
> **Note**   When you specify the **predictor route-map** command in SLB server farm configuration mode, no further commands in SLB server farm configuration mode or real server configuration mode are allowed.

For more information about policy-based routing, see the "Policy-Based Routing" and "Configuring Policy-Based Routing" sections of the *Cisco IOS IP Routing Configuration Guide* .

In a mobile Service Exchange Framework (mSEF) environment, Turbo RADIUS load balancing does not require firewall load balancing on the network side of the CSG cluster. (Standard RADIUS load balancing does require firewall load balancing on the network side of the cluster.)

Turbo RADIUS load balancing:

- • Supports simple IP access control lists (ACLs) and match and set next-hop pairs.
- • Is mutually exclusive with RADIUS load balancing maps and RADIUS load balancing accounting local acknowledgement.
- • Is mutually exclusive with the optional ACL logging facility. In order to use Turbo RADIUS load balancing, you must first disable the logging facility. For more information, see the description of the **access-list (IP standard)** command in the *Cisco IOS Security Command Reference* , Cisco IOS 12.4.

WAP Load Balancing

You can use IOS SLB to load-balance wireless session protocol (WSP) sessions among a group of WAP gateways or servers on an IP bearer network. WAP runs on top of UDP on a set of well known ports, with each port indicating a different WAP mode:

- Connectionless WSP mode (IP/UDP [9200]/WSP). In connectionless WSP mode, WSP is a simple 1-request/1-response protocol in which one server-bound packet results in a server response of one or more packets.
- Connection-oriented WSP mode (IP/UDP [9201]/WTP/WSP). In connection-oriented WSP mode, WTP manages retransmissions of WDP events, and WSP operates using a defined session bring-up/tear-down sequence. IOS SLB uses a WAP-aware finite state machine (FSM), driven by events in WSP sessions, to reassign sessions. This FSM operates only on port 9201, where the WSP sessions are not encrypted and WTP manages retransmissions.
- Connectionless secure WSP mode (IP/UDP [9202]/WTLS/WSP). This mode functions the same as connectionless WSP mode, but with security provided by WTLS.
- Connection-oriented secure WSP mode (IP/UDP [9203]/WTLS/WTP/WSP). This mode functions the same as connection-oriented WSP mode, but with security provided by WTLS.

IOS SLB uses WSP probes to detect failures in the WAP stack on port 9201.

Stateful Backup of Redundant Route Processors

When used with RPR+, IOS SLB supports the stateful backup of redundant route processors for mSEF for the CiscoCisco 7600 routers. This enables you to deploy Cisco Multiprocessor WAN Application Modules (MWAMs) in the same chassis as IOS SLB, while maintaining high availability of load-balancing assignments.

Flow Persistence

Flow persistence provides intelligent return routing of load-balanced IP flows to the appropriate node, without the need for coordinated hash mechanisms on both sides of the load-balanced data path, and without using Network Address Translation (NAT) or proxies to change client or server IP addresses.

Restrictions for Cisco IOS SLB

General Restrictions

- *Does not support load balancing of flows between clients and real servers that are on the same local-area network (LAN) or virtual LAN (VLAN). The packets being load-balanced cannot enter and leave the load-balancing device on the same interface.*
- You cannot configure IOS SLB from different user sessions at the same time.
- Do not configure an IOS SLB virtual IP address on the same subnet as any real server IP address, unless all server farms that include the real server IP address are configured using the **nat server** command.
- Operates in a standalone mode and currently does not operate as a MultiNode Load Balancing (MNLB) Services Manager. Does not support IOS SLB and MNLB configured with the same virtual IP address, even if they are for different services. The presence of IOS SLB does not preclude the use of the existing MNLB Forwarding Agent with an external Services Manager (such as the LocalDirector) in an MNLB environment. (MNLB is sometimes called Cisco Application Services Architecture, or CASA.)
- Does not support coordinating server load-balancing statistics among different IOS SLB instances for backup capability.
- Supports FTP and firewall load balancing only in dispatched mode.
- Does not support Dynamic Host Configuration Protocol (DHCP) load balancing.

- Does not support Internet Protocol version 6 (IPv6).
- When operating in dispatched mode, real servers must be Layer 2-adjacent, tag-switched, or through a GRE tunnel.

When operating in directed mode with server NAT, real servers need not be Layer 2-adjacent to IOS SLB. This function allows for more flexible network design, because servers can be placed several Layer 3 hops away from the IOS SLB switch.

- When operating in directed mode as a member of a multicast group, IOS SLB can receive multicast flows but cannot send multicast flows. This is not a restriction when operating in dispatched mode.
- Supports client Network Address Translation (NAT) and server port translation for TCP and UDP virtual servers only.
- When balancing streams to a virtual IP address that is the same as one of the IOS interface IP addresses (loopback, Ethernet, and so on), IOS SLB treats all UDP packets to that address as traceroute packets and replies with "host unreachable" ICMP packets. This occurs even if the IOS listens to the target UDP port. To avoid this issue, configure the virtual server as a network (address/31), not as a host (address/32).
- Do not use the virtual IP address configured in the IOS SLB virtual server for UDP-based router management applications such as SNMP. Doing so can result in high CPU usage. (This is not a problem for a UDP virtual server that is configured with destination port number 0.)
- The DFP agent requires a delay between hello messages of at least 3 seconds. Therefore, if your DFP manager provides a timeout specification, you must set the timeout to at least 3 seconds.
- When both IOS SLB and the Web Cache Communication Protocol (WCCP) are configured on a Cisco Catalyst 6500 series switch, and WCCP Input Redirection is configured with IOS SLB, Layer 2 WCCP forwarding must be used between the router and the cache. In this case, WCCP and IOS SLB both run in hardware and are processed in the correct order. If Generic Routing Encapsulation (GRE) forwarding is used, then IOS SLB takes precedence over WCCP and there is no redirection, because GRE forwarding is done on the MSFC. Note that the WCCP forwarding method, either Layer 2 or GRE, is configured on the cache engine and not on the switch.
- Do not configure IOS SLB and a Cisco Service Selection Gateway (SSG) on the same device.
- For "sandwich" configurations (that is, those that require an IOS SLB on both sides of a farm of CSGs, SSGs, or firewalls), if a flow is to be directed through two IOS SLB instances (virtual servers or firewall farms), the IOS SLB instances must reside in different Virtual Private Network (VPN) routing and forwardings (VRFs).
- If you do not configure an access interface using the **access** command in server farm, virtual server, or firewall farm configuration mode, IOS SLB installs the wildcards for the server farm, virtual server, or firewall farm in all of the available interfaces of the device, including the VRF interfaces. If IOS SLB is not required on the VRF interfaces, use the **access** command to limit wildcards to the specified interfaces only.
- VRF-aware IOS SLB does not operate "between" VRFs. That is, the server farm interface and the client traffic interface must use the same VRFs.

Static NAT Restrictions

- Does not work with client NAT server farms. That is, if a real server is using a virtual IP address for server NAT, and a server farm is associated with that same virtual IP address, then you cannot configure the server farm to use client NAT.
- Requires that each real server be associated with only one virtual server, to ensure that IOS SLB can create connections correctly.
- Requires a 0-port virtual server.
- Does not support virtual service FTP.
- Static NAT with per-packet server load balancing does not load-balance fragmented packets.

Backup Server Farm Restrictions

- Does not support defining the same real server in both primary and backup server farms.
- Requires the same NAT configuration (none, client, server, or both) for both primary and backup server farms. In addition, if NAT is specified, both server farms must use the same NAT pool.
- Does not support HTTP redirect load balancing. If a primary server farm specifies a redirect virtual server, you cannot define that primary as a backup, nor can you define a backup for that primary.

Firewall Load Balancing Restrictions

- Is not limited to one firewall farm in each load-balancing device.
- Requires that each firewall must have its own unique MAC address and must be Layer 2-adjacent to each device. The firewalls can be connected to individual interfaces on the device, or they can all share a VLAN and connect using one interface.
- Requires an Ethernet interface between each firewall load-balancing device and each firewall.
- On each firewall load-balancing device, IOS SLB requires that each Layer 2 firewall be connected to one Layer 3 (IP) interface.
- Flows with a destination IP address on the same subnet as the configured firewall IP addresses are not load-balanced. (Such flows could be a firewall console session or other flows on the firewall LAN.)
- Does not support the following IOS SLB functions:

  - NAT
  - Port-bound servers
  - SynGuard
  - TCP session reassignment
  - Transparent web cache load balancing

Restrictions for GPRS Load Balancing Without GPRS Tunneling Protocol (GTP) Cause Code Inspection Enabled

- If a real server is defined in two or more server farms, each server farm must be associated with a different virtual server.
- Operates in either dispatched or directed server NAT mode only.
- Supports stateful backup only if sticky connections are enabled.
- Cannot load-balance network-initiated PDP context requests.
- Does not support the following IOS SLB functions:

  - Bind IDs (A bind ID allows one physical server to be bound to multiple virtual servers and report a different weight for each.)
  - Client-assigned load balancing
  - Slow start
  - Weighted least connections load-balancing algorithm

Restrictions for GPRS Load Balancing With GTP Cause Code Inspection Enabled

- If a real server is defined in two or more server farms, each server farm must be associated with a different virtual server.
- Operates in directed server NAT mode only.
- Cannot load-balance network-initiated PDP context requests.
- Requires inbound and outbound signaling to flow through IOS SLB.
- Requires either the SGSN or the GGSN to echo its peer.

- Does not support the following IOS SLB functions:

  ◦ Bind IDs
  ◦ Client-assigned load balancing
  ◦ Slow start

GTP v2 Restrictions

- Does not support client NAT.
- IOS SLB can balance GTP v2 control packets for packet data network gateways (PGWs) and for serving gateways (SGWs). If a PGW load-balancing device and an SGW load-balancing device are configured in the same Supervisor Engine, you must configure a separate virtual server for each device.
- IOS SLB checks for and processes only the following GTP v2 messages.:

  ◦ GTP_CREATE_SESSION_REQ
  ◦ GTP_ECHO_REQ
  ◦ GTP_SLB_NOTIFICATION

All other messages are dropped.

- IOS SLB supports the following GTP_SLB notification messages:

  ◦ GTP_SLB_NOTIF_REASSIGN_REAL
  ◦ GTP_SLB_NOTIF_PDP_DELETION.
  ◦ GTP_SLB_NOTIF_PDP_STATUS

VPN Server Load Balancing Restrictions

- Does not support Internet Control Message Protocol (ICMP) and wildcard (0-protocol) virtual servers.

RADIUS Load Balancing Accelerated Data Plane Forwarding Restrictions

- Requires the route map algorithm.
- Requires redundant CSGs for best results.
- Requires static provisioning of load distribution by subscriber address range.
- Supports only simple IP access control lists (ACLs).
- When VSA correlation is used, IOS SLB maintains the correlation information only in the active RADIUS load-balancing device, not in the backup RADIUS load-balancing device. The backup RADIUS load-balancing device does not receive VSA correlation information from the active RADIUS load-balancing device.
- All Accounting-Request and Access-Accept messages must include the RADIUS-assigned Framed-ip-address attribute. The source IP address for each subscriber flow must also match the value of the Framed-ip-address attribute in the Access-Accept message.
- RADIUS accounting must be enabled on the RADIUS client, which is typically a Network Access Server (NAS).
- When you specify the **predictor route-map** command in SLB server farm configuration mode, no further commands in SLB server farm configuration mode or real server configuration mode are allowed.

VSA Correlation Restrictions

- VSA correlation might result in a degradation of performance.

- IOS SLB maintains the correlation information only in the active RADIUS load-balancing device, not in the backup RADIUS load-balancing device. The backup RADIUS load-balancing device does not receive VSA correlation information from the active RADIUS load-balancing device.
- The Cisco VSA is injected into the RADIUS Accounting-Start packet. The Cisco VSA is not injected into any other RADIUS messages or packets, such as interim RADIUS Accounting On or Off messages or RADIUS Accounting-Stop packets.
- You cannot configure **radius inject acct** commands and **radius inject auth** commands on the same virtual server.

RADIUS Load Balancing for GPRS Restrictions

- Requires the weighted round robin algorithm.
- Does not support fragmented RADIUS packets.
- All Accounting-Request and Access-Accept messages must include the RADIUS-assigned Framed-ip-address attribute. The source IP address for each subscriber flow must also match the value of the Framed-ip-address attribute in the Access-Accept message.
- RADIUS accounting must be enabled on the RADIUS client, which is typically a Network Access Server (NAS).

RADIUS Load Balancing for CDMA2000 Restrictions

- Requires the weighted round robin algorithm.
- Does not support fragmented RADIUS packets.
- All subscribers on the mobile network must be assigned a unique IP address (that is, no overlapping IP addresses) which can be routed in the mobile wireless network.
- Each User-Name attribute must correspond to one subscriber, or at most to a very small number of subscribers. Otherwise, one SSG might be burdened with an unexpectedly large load.
- For simple IP networks, the following additional restrictions apply:

  ◦ The PDSN must include the User-Name attribute in all RADIUS Access-Request and Accounting-Start packets. The value of the User-Name attribute for a subscriber must be the same in all the packets (except for Cisco PDSNs that provide MSID-based access).
  ◦ The PDSN must include the Framed-ip-address attribute and the NAS-ip-address in all RADIUS Accounting-Start and Accounting-Stop packets. The value of the Framed-ip-address attribute must equal the source IP address in subscriber data packets routed by RADIUS load balancing for SSG service.
  ◦ The PDSN must include the NAS-ip-address in all Accounting-Requests. For BSC/PCF hand-offs, the Accounting-Stop must include the 3GPP2-Session-Continue VSA with a value of **1**, to prevent the destruction of RADIUS load balancing sticky database objects for the subscriber.
- For Mobile IP networks, the following additional restrictions apply:

  ◦ For a subscriber session, the PDSN and HA must send the RADIUS Access-Request and Accounting-Start packets with the User-Name attribute. The value of the User-Name attribute in all PDSN and HA RADIUS packets must be the same for the session.
  ◦ For a subscriber session, the PDSN and HA must send RADIUS Accounting-Request packets with a Framed-ip-address attribute equal to the source IP address in subscriber data packets routed by RADIUS load balancing for SSG service. All RADIUS Accounting-Requests sent by the PDSN and HA must also include the NAS-ip-address attribute.
  ◦ The PDSN must include the 3GPP2-Correlation-Identifier attribute in all Accounting-Requests.

Home Agent Director Restrictions

- A Registration Request (RRQ) must include the network access identifier (NAI) to be load-balanced.
- An RRQ must include a home agent IP address of either 0.0.0.0 or 255.255.255.255 to be load-balanced.
- For fast switching, the NAI in the RRQ cannot be more than 96 bytes deep in the packet. If the NAI is deeper than 96 bytes, IOS SLB manages the packet at the process level.
- Operates in either dispatched or directed server NAT mode only.
- Does not support the following IOS SLB functions:

  - Bind IDs
  - Client-assigned load balancing
  - Slow start
  - Stateful backup
  - Sticky connections
  - Weighted least connections load-balancing algorithm

Restrictions for HTTP Probes

- HTTP probes do not support HTTP over Secure Socket Layer (HTTPS). That is, you cannot send an HTTP probe to an SSL server.

Restrictions for UDP Probes

- UDP probes do not support fragmented Response packets.
- UDP probes do not support hosts that require a particular source port value in probe packets. UDP probes select an ephemeral port for each probe.
- Protocols and applications that have Message Digest Algorithm Version 5 (MD5) checksums generated from payload must be captured by a "sniffer" to obtain a correct checksum.
- For Cisco IOS Multiprotocol Label Switching (MPLS):

  - Clients can connect to IOS SLB through the MPLS cloud in a Supervisor Engine 720 environment.
  - The MPLS client interface must be configured with Tunnel Engineering. No other MPLS configuration is supported.
  - The MPLS client interface must receive packets as IP packets.
  - The MPLS client interface must be behind a Penultimate Hop Popping (PHP) router.
- For Cisco Catalyst 6500 series switches and Cisco 7600 series routers:

  - Supports Native Cisco IOS only (c6sup images). Native Cisco IOS requires the MSFC and the Policy Feature Card (PFC). When running redundant MSFCs in the same Catalyst 6500 switch, stateful backup between the two MSFCs is not supported, but stateless backup between the two MSFCs is supported.

The term MSFC refers to an MSFC1, MSFC2, or MSFC3, except when specifically differentiated.

The term PFC refers to a PFC1, PFC2, or PFC3, except when specifically differentiated.

- - Requires that the Multilayer Switching (MLS) flow mode operate in full-flow mode or in interface full-flow mode. IOS SLB automatically sets the flow mode for its own use. For more information about how to set the MLS flow, refer to the *Catalyst 6000 Family Cisco IOS Software Configuration Guide* .
  - When operating in dispatched mode, real servers must be Layer 2-adjacent to IOS SLB (that is, not beyond an additional router), with hardware data packet acceleration performed by the PFC.

All real servers in the same server farm must be on the same VLAN. The loopback address must be configured in the real servers.

◦ Requires that all real servers in a firewall farm be on the same VLAN. Real servers in different firewall farms can be on different VLANs.

◦ Provides no hardware data packet acceleration in directed mode. (Hardware data packet acceleration is performed by the PFC, and in directed mode the packets are managed by the MSFC, not the PFC.)

◦ For the Cisco Supervisor Engine 2, "sandwich" configurations that require firewall load balancing are not supported, because such configurations require VRF. VRF is not supported for the Supervisor Engine 2.

ASN Release 6 Load Balancing Restrictions

- Operates in either dispatched or directed server NAT mode only. In directed mode, IOS SLB changes the destination IP address of the Mobile Station Pre-Attachment request to that of the selected Access Service Network (ASN) gateway real server.
- Requires DFP
- Does not support the following features:

  ◦ Client NAT
  ◦ Weighted least connections algorithm (for Mobile Station Pre-Attachment requests)

- When the base station is configured to send Mobile Station Pre-Attachment ACKnowledgement, or ACK, packets directly to an ASN gateway, bypassing IOS SLB, you must ensure that the session can time out without failing the real server. To do so, configure the **no faildetect inband** command real server configuration mode.
- For stateful backup and sticky connections:

  ◦ ASN sticky connections are supported only on the Cisco Broadband Wireless Gateway (BWG) Release 2.0 or later.
  ◦ If you are running ASN on a Cisco BWG, we recommend that you configure the **gw port** command in virtual server configuration mode.
  ◦ Do not use port number 2231 as the communication port between the Cisco BWG and the IOS SLB that is providing load balancing for the ASN.
  ◦ If you are not running ASN on a Cisco BWG, you must use the **sticky**command in virtual server configuration mod for sticky object deletion, since delete and network address identifier (NAI) update notifications on communication ports are not expected.
  ◦ To enable the Cisco BWG to send notifications for ASN to IOS SLB, configure the **wimax agw slb port** command in global configuration mode on the Cisco BWG.

Note Cisco BWG commands are documented in the *Cisco Broadband Wireless Gateway Command Reference* .

- ◦ To enable the Cisco BWG to send NAI-update notifications to IOS SLB when an MSID is registered, configure the **wimax agw slb notify nai-updates** command in global configuration mode on the Cisco BWG.
  ◦ To enable the Cisco BWG to send delete notifications to IOS SLB when an MSID is unregistered or deleted, configure the **wimax agw slb notify session-deletion** command in global configuration mode on the Cisco BWG.When you configure IOS SLB firewall load balancing, the load-balancing devices use route lookup to recognize flows destined for the firewalls. To enable route lookup, you must configure each device with the IP address of each firewall that will route flows to that device.

How to Configure IOS SLB Features

Configuring IOS SLB involves identifying server farms, configuring groups of real servers in server farms, and configuring the virtual servers that represent the real servers to the clients.

For configuration examples associated with these tasks, see the "Configuration Examples for IOS SLB" section.

For a complete description of the IOS SLB commands in this section, refer to the "Server Load Balancing Commands" chapter of the *Cisco IOS IP Application Services Command Reference*. To locate documentation of other commands that appear in this section, search online using Cisco.com.

How to Configure Required and Optional IOS SLB Functions

To configure IOS SLB functions, perform the tasks in the following sections. Required and optional tasks are indicated.

How to Configure a Server Farm and a Real Server

Perform this required task to configure a server farm and a real server.

✎

Note You cannot configure IOS SLB from different user sessions at the same time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb serverfarm** *server-farm*
4. **access** *interface*
5. **bindid** [*bind-id*]
6. **nat** {**client** *pool* | **server**}
7. **predictor** [**roundrobin**| **leastconns**| **route-map** *mapname*]
8. **probe** *probe*
9. **real** *ipv4-address* [**ipv6** *ipv6-address*] [*port*]
10. **faildetect numconns** *number-of-conns* [**numclients** *number-of-clients*]
11. **maxclients** *number-of-conns*
12. **maxconns** *number-of-conns* [**sticky-override**]
13. **reassign** *threshold*
14. **retry** *retry-value*
15. **weight** *setting*
16. **inservice**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. If prompted, enter your password.

	Command or Action	Purpose
	Example: `Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip slb serverfarm** *server-farm* **Example:** `Router(config)# ip slb serverfarm PUBLIC`	Adds a server farm definition to the IOS SLB configuration and enters server farm configuration mode.
Step 4	**access** *interface* **Example:** `Router(config-slb-sfarm)# access GigabitEthernet 0/1.1`	(Optional) Configures an access interface or subinterface for a server farm.
Step 5	**bindid** [*bind-id*] **Example:** `Router(config-slb-sfarm)# bindid 309`	(Optional) Specifies a bind ID on the server farm for use by Dynamic Feedback Protocol (DFP). **Note** GPRS load balancing and Home Agent Director do not support this command.
Step 6	**nat** {**client** *pool* | **server**} **Example:** `Router(config-slb-sfarm)# nat server`	(Optional) Configures Network Address Translation (NAT) client translation mode or NAT server address translation mode on the server farm. All IPv4 or IPv6 server farms that are associated with the same virtual server must have the same NAT configuration.
Step 7	**predictor** [**roundrobin**| **leastconns**| **route-map** *mapname*] **Example:** `Router(config-slb-sfarm)# predictor leastconns`	(Optional) Specifies the algorithm to be used to determine how a real server is selected. **Note** RADIUS load balancing requires the default setting (the weighted round robin algorithm). In GPRS load balancing without GTP cause code inspection enabled, you must accept the default setting (the weighted round robin algorithm). The Home Agent Director requires the default setting (the weighted round robin algorithm). When you specify the **predictor route-map** command in SLB server farm configuration mode, no further commands in SLB server farm configuration mode or real server configuration mode are allowed. For more details, see the following sections: • "Weighted Round Robin Algorithm" • "Weighted Least Connections Algorithm" • "Route Map Algorithm"

	Command or Action	Purpose
Step 8	**probe** *probe* **Example:** `Router(config-slb-sfarm)# probe PROBE1`	(Optional) Associates a probe with the real server.
Step 9	**real** *ipv4-address* [**ipv6** *ipv6-address*] [*port*] **Example:** `Router(config-slb-sfarm)# real 10.1.1.1`	Identifies a real server by IPv4 address, and optional IPv6 address and port number, as a member of a server farm and enters real server configuration mode. **Note** In GPRS load balancing, specify the IP addresses (virtual template addresses, for Cisco GGSNs) of the real servers performing the GGSN function. In VPN server load balancing, specify the IP addresses of the real servers acting as VPN terminators. For the Home Agent Director, specify the IP addresses of the real servers acting as home agents. For dual-stack support for GTP load balancing, specify the real server's IPv4 and IPv6 address.
Step 10	**faildetect numconns** *number-of-conns* [**numclients** *number-of-clients*] **Example:** `Router(config-slb-real)# faildetect numconns 10 numclients 3`	(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 GPRS load balancing, if only one SGSN is configured in your environment, specify the **numclients** keyword with a value of 1. • In RADIUS load balancing, for automatic session-based failure detection, specify the **numclients** keyword with a value of 1.
Step 11	**maxclients** *number-of-conns* **Example:** `Router(config-slb-real)# maxclients 10`	(Optional) Specifies the maximum number of IOS SLB RADIUS and GTP sticky subscribers that can be assigned to an individual virtual server.
Step 12	**maxconns** *number-of-conns* [**sticky-override**] **Example:** `Router(config-slb-real)# maxconns 1000`	(Optional) Specifies the maximum number of active connections allowed on the real server at one time.
Step 13	**reassign** *threshold* **Example:** `Router(config-slb-real)# reassign 2`	(Optional) Specifies the threshold of consecutive unacknowledged SYNchronize sequence numbers (SYNs) or Create Packet Data Protocol (PDP) requests that, if exceeded, result in an attempted connection to a different real server. **Note** In GPRS load balancing, you must specify a reassign threshold less than the SGSN's N3-REQUESTS counter value.
Step 14	**retry** *retry-value* **Example:** `Router(config-slb-real)# retry 120`	(Optional) Specifies the time interval, in seconds, to wait between the detection of a server failure and the next attempt to connect to the failed server.

	Command or Action	Purpose
Step 15	**weight** *setting* **Example:** Router(config-slb-real)# weight 24	(Optional) Specifies the real server workload capacity relative to other servers in the server farm. **Note** If you use Dynamic Feedback Protocol (DFP), the static weights you define using the weight command in server farm configuration mode are overridden by the weights calculated by DFP. If DFP is removed from the network, IOS SLB reverts to the static weights.
Step 16	**inservice** **Example:** Router(config-slb-real)# inservice	Enables the real server for use by IOS SLB.

Note   When performing server load balancing and firewall load balancing together on a Cisco Catalyst 6500 Family Switch, use the **mls ip slb wildcard search rp** command to reduce the probability of exceeding the capacity of the Telecommunications Access Method (TCAM) on the Policy Feature Card (PFC). See "How to Configure a Wildcard Search" for more details.

How to Configure a Virtual Server

Perform this required task to configure a virtual server. IOS SLB supports up to 500 virtual servers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb vserver** *virtual-server*
4. Do one of the following:

    - **virtual** *ipv4-address* [*ipv4-netmask*[**group**]] {**esp**| **gre**| *protocol*}
    -
    -
    - **virtual** *ipv4-address* [*ipv4-netmask*[**group**]] [**ipv6** *ipv6-address* [**prefix** *ipv6-prefix*]] {**tcp**| **udp**} [*port*| **any**] [**service** *service*]

5. **serverfarm** *primary-farm* [**backup** *backup-farm*[**sticky**]] [**ipv6-primary** *ipv6-primary-farm*[**ipv6-backup** *ipv6-backup-farm*]] [**map** *map-id* **priority** *priority*]
6. **access** *interface* [**route framed-ip**]
7. **advertise** [**active**]
8. **client** {*ipv4-address netmask*[**exclude**] | **gtp carrier-code** [*code*]}
9. **delay** {*duration* | **radius framed-ip** *duration*}
10. **gtp notification cac** [*reassign-count*]
11. **gtp session**
12. **gw port** *port*
13. **hand-off radius** *duration*
14. **idle** [**asn request** *duration* | **asn msid** *msid* | **gtp imsi** *duration* [**query** [*max-queries*]] | **gtp request** *duration* | **ipmobile request** *duration* | **radius** {**request** | **framed-ip**} *duration*]
15. **purge radius framed-ip acct on-off**
16. **purge radius framed-ip acct stop** {*attribute-number* | {**26**| *vsa*} {*vendor-ID* | **3gpp**| **3gpp2**} *sub-attribute-number*}
17. **radius acct local-ack key** [*encrypt*] *secret-string*
18. **radius inject auth** *group-number* {**calling-station-id** | **username**}
19. **radius inject auth timer** *seconds*
20. **radius inject auth vsa** *vendor-id*
21. **replicate casa** *listen-ip remote-ip port* [*interval*] [**password** [*encrypt*] *secret-string timeout*]
22. **replicate interval** *interval*
23. **replicate slave**
24. **sticky** {*duration*[**group** *group-id*] [**netmask** *netmask*] | **asn msid**[**group** *group-id*] | **gtp imsi**[**group** *group-id*] | **radius calling-station-id**| **radius framed-ip**[**group** *group-id*] | **radius username**[**msid-cisco**] [**group** *group-id*]}
25. **synguard** *syn-count interval*
26. **inservice** [**standby** *group-name*] [**active**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. If prompted, enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**ip slb vserver** *virtual-server*
	
	Example:
	
	`Router(config)# ip slb vserver`
	`PUBLIC_HTTP`
		Identifies a virtual server and enters virtual server configuration mode.

Step 4	Do one of the following:
	
	• **virtual** *ipv4-address* [*ipv4-netmask*[**group**]] {**esp**| **gre**| *protocol*}
	
	•
	
	•
	
	• **virtual** *ipv4-address* [*ipv4-netmask*[**group**]] [**ipv6** *ipv6-address* [**prefix** *ipv6-prefix*]] {**tcp**| **udp**} [*port*| **any**] [**service** *service*]
	
	Example:
	
	`Router(config-slb-vserver)# virtual`
	`10.0.0.1 tcp www`
		Specifies the virtual server IP address, type of connection, and optional TCP or User Datagram Protocol (UDP) port number, Internet Key Exchange (IKE) or Wireless Session Protocol (WSP) setting, and service coupling.
		
		Note For RADIUS load balancing, specify the **service radius**keyword option.
		Note For ASN load balancing, specify the **service asn**keyword option.
		Note For GPRS load balancing:
		
		• ◦ Specify a virtual GGSN IP address as the virtual server, and specify the **udp** keyword option.
		◦ To load-balance GTP v1 and GTP v2 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 GPRS load balancing *without* GTP cause code inspection, specify the **service gtp**keyword option.
		◦ To enable GPRS load balancing *with* GTP cause code inspection, specify the **service gtp-inspect**keyword option.
		◦ For dual-stack support for GTP load balancing, specify the virtual server's IPv4 and IPv6 addresses and optional IPv6 prefix.

Step 5	**serverfarm** *primary-farm* [**backup** *backup-farm*[**sticky**]] [**ipv6-primary** *ipv6-primary-*	Associates a real server farm with a virtual server, and optionally configures a backup server farm and specifies that sticky connections are to be used in the backup server farm.

Command or Action	Purpose
farm[**ipv6-backup** *ipv6-backup-farm*]] [**map** *map-id* **priority** *priority*] **Example:** Router(config-slb-vserver)# serverfarm SF1 backup SF2 map 1 priority 1	**Note**RADIUS load balancing and the Home Agent Director do not support the **sticky**keyword. You can associate more than one server farm with a given RADIUS virtual server by configuring more than one **serverfarm** command, each with a unique map ID and a unique priority. (That is, each map ID and each map priority must be unique across all server farms associated with the virtual server.) For GPRS load balancing, if a real server is defined in two or more server farms, each server farm must be associated with a different virtual server. For dual-stack support for GTP load balancing, specify the primary IPv6 server farm and optional backup IPv6 server farm. All IPv4 or IPv6 server farms that are associated with the same virtual server must have the same NAT configuration.
Step 6 **access** *interface* [**route framed-ip**] **Example:** Router(config-slb-vserver)# access Vlan20 route framed-ip	(Optional) Enables framed-IP routing to inspect the ingress interface.
Step 7 **advertise** [**active**] **Example:** Router(config-slb-vserver)# advertise	(Optional) Controls the installation of a static route to the Null0 interface for a virtual server address.
Step 8 **client** {*ipv4-address netmask*[**exclude**] | **gtp carrier-code** [*code*]} **Example:** Router(config-slb-vserver)# client 10.4.4.0 255.255.255.0	(Optional) Specifies which clients are allowed to use the virtual server. **Note**GPRS load balancing supports only the **gtp carrier-code** option, and only if GTP cause code inspection is enabled. Dual-stack support for GTP load balancing does not support this command.
Step 9 **delay** {*duration* | **radius framed-ip** *duration*} **Example:** Router(config-slb-vserver)# delay 30	(Optional) Specifies the time IOS SLB maintains TCP connection context after a connection has ended.
Step 10 **gtp notification cac** [*reassign-count*] **Example:** Router(config-slb-vserver)# gtp notification cac 5	(Optional) Limits the number of times IOS SLB can reassign a session to a new real server for GGSN-IOS SLB messaging.
Step 11 **gtp session** **Example:** Router(config-slb-vserver)# no gtp session	(Optional) Enables IOS SLB to create GTP load-balancing sessions. This is the default setting. To enable sticky-only load balancing for GTP, use the **no** form of this command: **no gtp session**

	Command or Action	Purpose
		If you enable sticky-only load balancing, you must also enable sticky connections for the virtual server using the **sticky (virtual server)**command.
Step 12	**gw port** *port* **Example:** `Router(config-slb-vserver)# gw port 63082`	(Optional) Specifies the port that the Cisco Broadband Wireless Gateway (BWG) is to use to communicate with IOS SLB.
Step 13	**hand-off radius** *duration* **Example:** `Router(config-slb-vserver)# hand-off radius 30`	(Optional) Changes the amount of time IOS SLB waits for an ACCT-START message from a new Mobile IP foreign agent in the event of a foreign agent hand-off.
Step 14	**idle** [**asn request** *duration* | **asn msid** *msid* | **gtp imsi** *duration* [**query** [*max-queries*]] | **gtp request** *duration* | **ipmobile request** *duration* | **radius** {**request** | **framed-ip**} *duration*] **Example:** `Router(config-slb-vserver)# idle 120`	(Optional) Specifies the minimum time IOS SLB maintains connection context in the absence of packet activity. **Note**In GPRS load balancing *without* GTP cause code inspection enabled, specify an idle timer greater than the longest possible interval between PDP context requests on the SGSN.
Step 15	**purge radius framed-ip acct on-off** **Example:** `Router(config-slb-vserver)# purge radius framed-ip acct on-off`	(Optional) Enables IOS SLB to purge entries in the IOS SLB RADIUS framed-IP sticky database upon receipt of an Accounting On or Off message.
Step 16	**purge radius framed-ip acct stop** {*attribute-number* | {**26**| *vsa*} {*vendor-ID* | **3gpp**| **3gpp2**} *sub-attribute-number*} **Example:** `Router(config-slb-vserver)# purge radius framed-ip acct stop 44`	(Optional) Enables IOS SLB to purge entries in the IOS SLB RADIUS framed-IP sticky database upon receipt of an Accounting-Stop message.
Step 17	**radius acct local-ack key** [*encrypt*] *secret-string* **Example:** `Router(config-slb-vserver)# radius acct local-ack key SECRET_PASSWORD`	(Optional) Enables a RADIUS virtual server to acknowledge RADIUS accounting messages.
Step 18	**radius inject auth** *group-number* {**calling-station-id** | **username**}	(Optional) Configures a vendor-specific attribute (VSA) correlation group for an IOS SLB RADIUS load balancing accelerated data plane forwarding authentication virtual server, and specifies whether IOS SLB is to create VSA correlation entries based on RADIUS calling station IDs or RADIUS usernames.

	Command or Action	Purpose

	Example:
	
	`Router(config-slb-vserver)# radius`
	`inject auth 1 calling-station-id`	

Step 19	**radius inject auth timer** *seconds*
	
	Example:
	
	`Router(config-slb-vserver)# radius`
	`inject auth timer 45`
		(Optional) Configures a timer for VSA correlation for an IOS SLB RADIUS load balancing accelerated data plane forwarding authentication virtual server.

Step 20	**radius inject auth vsa** *vendor-id*
	
	Example:
	
	`Router(config-slb-vserver)# radius`
	`inject auth vsa vendor1`
		(Optional) Buffers VSAs for VSA correlation for an IOS SLB RADIUS load balancing accelerated data plane forwarding authentication virtual server.

Step 21	**replicate casa** *listen-ip remote-ip port* [*interval*] [**password** [*encrypt*] *secret-string timeout*]
	
	Example:
	
	`Router(config-slb-vserver)# replicate`
	`casa 10.10.10.11 10.10.11.12 4231`
		(Optional) Configures a stateful backup of IOS SLB decision tables to a backup switch.
		
		Note The Home Agent Director does not support this command. If you specify the **service gtp** keyword on the **virtual** command, and you do not specify the **sticky** command with the **gtp imsi** keyword, the **replicate casa** command is not supported (because sessions are not persistent, and there is nothing to replicate).

Step 22	**replicate interval** *interval*
	
	Example:
	
	`Router(config-slb-vserver)# replicate`
	`interval 20`
		(Optional) Sets the replication delivery interval for an IOS SLB virtual server.
		
		Note The Home Agent Director does not support this command. If you specify the **service gtp** keyword on the **virtual** command, and you do not specify the **sticky** command with the **gtp imsi** keyword, the **replicate casa** command is not supported (because sessions are not persistent, and there is nothing to replicate).

Step 23	**replicate slave**
	
	Example:
	
	`Router(config-slb-vserver)# replicate`
	`slave`
		(Optional) Enables stateful backup of redundant route processors for an IOS SLB virtual server.
		
		Note The Home Agent Director does not support this command. If you specify the **service gtp** keyword on the **virtual** command, and you do not specify the **sticky** command with the **gtp imsi** keyword, the **replicate casa** command is not supported (because sessions are not persistent, and there is nothing to replicate). If you are using one Supervisor Engine with **replicate slave** configured, you might receive out-of-sync messages on the Supervisor.

Step 24	**sticky** {*duration*[**group** *group-id*] [**netmask** *netmask*]	**asn msid**[**group** *group-id*]	**gtp imsi**[**group** *group-id*]	**radius calling-station-id**| **radius framed-ip**[**group** *group-id*]	**radius username**[**msid-cisco**] [**group** *group-id*]}	(Optional) Specifies that connections from the same client use the same real server, as long as the interval between client connections does not exceed the specified duration. **Note** In VPN server load balancing, specify a *duration* of at least 15 seconds. GPRS load balancing and the Home Agent Director do not support this command.

	Command or Action	Purpose
	Example: `Router(config-slb-vserver)# sticky 60 group 10`	
Step 25	**synguard** *syn-count interval* **Example:** `Router(config-slb-vserver)# synguard 50`	(Optional) Specifies the rate of TCP SYNchronize sequence numbers (SYNs) managed by a virtual server in order to prevent a SYN flood denial-of-service attack. **Note**GPRS load balancing and the Home Agent Director do not support this command.
Step 26	**inservice** [**standby** *group-name*] [**active**] **Example:** `Router(config-slb-vserver)# inservice`	Enables the virtual server for use by IOS SLB.

How to Verify a Virtual Server

Perform the following optional task to verify a virtual server.

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

```
Router# show ip slb vservers
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#
```

How to Verify a Server Farm

Perform the following optional task to verify a server farm.

The following **show ip slb reals** command shows 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#
```

How to Verify Clients

Perform the following optional task to verify 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 shows detailed information about the restricted client access status:

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

How to Verify IOS SLB Connectivity

Perform the following optional task to verify IOS SLB connectivity.

To verify that the IOS SLB feature is 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 shows detailed information about the IOS SLB network status:

```
Router# show ip slb stats
 Pkts via normal switching:  0
 Pkts via special switching: 6
 Pkts dropped:               0
 Connections Created:        1
 Connections Established:    1
 Connections Destroyed:      0
 Connections Reassigned:     0
 Zombie Count:               0
 Connections Reused:         0
```

- Normal switching exists when IOS SLB packets are managed on normal IOS switching paths (CEF, fast switching, and process level switching).
- Special switching exists when IOS SLB packets are managed on hardware-assisted switching paths.

See "How to Monitor and Maintain the Cisco IOS SLB Feature" for additional commands used to verify IOS SLB networks and connections.

How to Configure Firewall Load Balancing

Perform the following tasks to configure a basic IOS SLB firewall load-balancing network.

IOS SLB firewall load balancing uses probes to detect and recover from failures. You must configure a probe on each real server in the firewall farm. Ping probes are recommended; see "How to Configure a Ping Probe" for more details. If a firewall does not allow ping probes to be forwarded, use HTTP probes instead. See "How to Configure an HTTP Probe" for more details. You can configure more than one probe, in any combination of supported types (DNS, HTTP, TCP, or ping), for each firewall in a firewall farm.

When you perform server load balancing and firewall load balancing together on a Cisco Catalyst 6500 switch, use the **mls ip slb wildcard search rp** command in global configuration mode to reduce the

probability of exceeding the capacity of the Telecommunications Access Method (TCAM) on the Policy Feature Card (PFC). See "How to Configure a Wildcard Search" for more details.

If IOS SLB experiences a high purge rate, the CPU might be impacted. If this problem occurs, use the **no** form of the **mls ip slb purge global** command in global configuration mode to disable purge throttling on TCP and UDP flow packets. See "How to Configure Protocol-Level Purging of MLS Entries" for more details.

This section describes the following IOS SLB firewall load-balancing configuration tasks. Required and optional tasks are indicated.

How to Configure a Firewall Farm

Perform the following required task to configure a firewall farm.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb firewallfarm** *firewall-farm*
4. **real** *ip-address*
5. **probe** *probe*
6. **weight** *setting*
7. **inservice**
8. **access** [**source** *source-ip netmask* | **destination** *destination-ip netmask* | **inbound** {*inbound-interface* | **datagram connection**} | **outbound** *outbound-interface*]
9. **predictor hash address** [**port**]
10. **purge connection**
11. **purge sticky**
12. **replicate casa** *listen-ip remote-ip port* [*interval*] [**password** [*encrypt*] *secret-string*[*timeout*]]
13. **replicate interval** *interval*
14. **replicate slave**
15. **protocol tcp**
16. **delay** *duration*
17. **idle** *duration*
18. **maxconns** *maximum-number*
19. **sticky** *duration* [**netmask** *netmask*] [**source**| **destination**]
20. **protocol datagram**
21. **idle** *duration*
22. **maxconns** *maximum-number*
23. **sticky** *duration* [**netmask** *netmask*] [**source**| **destination**]
24. **inservice**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**ip slb firewallfarm** *firewall-farm* **Example:** Router(config)# ip slb firewallfarm FIRE1	Adds a firewall farm definition to the IOS SLB configuration and enters firewall farm configuration mode.
Step 4	**real** *ip-address* **Example:** Router(config-slb-fw)# real 10.1.1.1	Identifies a firewall by IP address as a member of a firewall farm and enters real server configuration mode.
Step 5	**probe** *probe* **Example:** Router(config-slb-fw-real)# probe FireProbe	Associates a probe with the firewall.
Step 6	**weight** *setting* **Example:** Router(config-slb-fw-real)# weight 24	(Optional) Specifies the firewall's workload capacity relative to other firewalls in the firewall farm.
Step 7	**inservice** **Example:** Router(config-slb-fw-real)# inservice	Enables the firewall for use by the firewall farm and by IOS SLB.
Step 8	**access** [**source** *source-ip netmask* | **destination** *destination-ip netmask* | **inbound** {*inbound-interface* | **datagram connection**} | **outbound** *outbound-interface*]	(Optional) Routes specific flows to a firewall farm.

	Command or Action	Purpose

	Example:
	
	`Router(config-slb-fw)# access`
	`destination 10.1.6.0 255.255.255.0`	

Step 9	**predictor hash address** [**port**]
	
	Example:
	
	`Router(config-slb-fw)# predictor hash`
	`address`
		(Optional) Specifies whether the source and destination TCP or User Datagram Protocol (UDP) port numbers, in addition to the source and destination IP addresses, are to be used when selecting a firewall.

Step 10	**purge connection**
	
	Example:
	
	`Router(config-slb-fw)# purge`
	`connection`
		(Optional) Enables IOS SLB firewall load balancing to send purge requests for connections.

Step 11	**purge sticky** **Example:** `Router(config-slb-fw)# purge sticky`	(Optional) Enables IOS SLB firewall load balancing to send purge requests for sticky connections when the sticky timer expires.

Step 12	**replicate casa** *listen-ip remote-ip port* [*interval*] [**password** [*encrypt*] *secret-string*[*timeout*]]
	
	Example:
	
	`Router(config-slb-fw)# replicate casa`
	`10.10.10.11 10.10.11.12 4231`
		(Optional) Configures a stateful backup of IOS SLB firewall load-balancing decision tables to a backup switch.
		
		Note The Home Agent Director does not support this command. If you specify the **service gtp**keyword on the **virtual** command, and you do not specify the **sticky**command with the **gtp imsi**keyword, the **replicate casa**command is not supported (because sessions are not persistent, and there is nothing to replicate).

Step 13	**replicate interval** *interval*
	
	Example:
	
	`Router(config-slb-fw)# replicate`
	`interval 20`
		(Optional) Sets the replication delivery interval for an IOS SLB firewall farm.
		
		Note The Home Agent Director does not support this command. If you specify the **service gtp**keyword on the **virtual** command, and you do not specify the **sticky**command with the **gtp imsi**keyword, the **replicate interval**command is not supported (because sessions are not persistent, and there is nothing to replicate).

Step 14	**replicate slave** **Example:** `Router(config-slb-fw)# replicate slave`	(Optional) Enables stateful backup of redundant route processors for an IOS SLB firewall farm. **Note** The Home Agent Director does not support this command. If you specify the **service gtp**keyword on the **virtual** command, and you do not specify the **sticky**command with the **gtp imsi**keyword, the **replicate slave**command is not supported (because sessions are not persistent, and there is nothing to replicate). If you are using one Supervisor Engine with **replicate slave** configured, you might receive out-of-sync messages on the Supervisor.
Step 15	**protocol tcp**	(Optional) Enters firewall farm TCP protocol configuration mode.

Command or Action	Purpose
Example: `Router(config-slb-fw)# protocol tcp`	
Step 16 **delay** *duration* **Example:** `Router(config-slb-fw-tcp)# delay 30`	(Optional) In firewall farm TCP protocol configuration mode, specifies the time IOS SLB firewall load balancing maintains TCP connection context after a connection ends.
Step 17 **idle** *duration* **Example:** `Router(config-slb-fw-tcp)# idle 120`	(Optional) In firewall farm TCP protocol configuration mode, specifies the minimum time IOS SLB firewall load balancing maintains connection context in the absence of packet activity.
Step 18 **maxconns** *maximum-number* **Example:** `Router(config-slb-fw-tcp)# maxconns 1000`	(Optional) In firewall farm TCP protocol configuration mode, specifies the maximum number of active TCP connections allowed on the firewall farm at one time.
Step 19 **sticky** *duration* [**netmask** *netmask*] [**source**|**destination**] **Example:** `Router(config-slb-fw-tcp)# sticky 60`	(Optional) In firewall farm TCP protocol configuration mode, specifies that connections from the same IP address use the same firewall if either of the following conditions is met: • As long as any connection between the same pair of IP addresses exists (source and destination sticky). • For a period, defined by *duration* , after the last connection is destroyed.
Step 20 **protocol datagram** **Example:** `Router(config-slb-fw)# protocol datagram`	(Optional) Enters firewall farm datagram protocol configuration mode.
Step 21 **idle** *duration* **Example:** `Router(config-slb-fw-udp)# idle 120`	(Optional) In firewall farm datagram protocol configuration mode, specifies the minimum time IOS SLB firewall load balancing maintains connection context in the absence of packet activity.
Step 22 **maxconns** *maximum-number* **Example:** `Router(config-slb-fw-udp)# maxconns 1000`	(Optional) In firewall farm datagram protocol configuration mode, specifies the maximum number of active datagram connections allowed on the firewall farm at one time.
Step 23 **sticky** *duration* [**netmask** *netmask*] [**source**|**destination**]	(Optional) In firewall farm datagram protocol configuration mode, specifies that connections from the same IP address use the same firewall if either of the following conditions is met:

	Command or Action	Purpose
	Example: `Router(config-slb-fw-udp)# sticky 60`	• As long as any connection between the same pair of IP addresses exists (source and destination sticky). • For a period, defined by *duration* , after the last connection is destroyed.
Step 24	**inservice** **Example:** `Router(config-slb-fw)# inservice`	Enables the firewall farm for use by IOS SLB.

How to Verify a Firewall Farm

Perform the following optional task to verify a firewall farm.

The following **show ip slb reals** command shows the status of firewall farm FIRE1, the associated real servers, and the server status:

```
Router# show ip slb real
real                farm name      weight  state         conns
-------------------------------------------------------------
10.1.1.2            FIRE1          8       OPERATIONAL   0
10.1.2.2            FIRE1          8       OPERATIONAL   0
```

The following **show ip slb firewallfarm**command shows the configuration and status of firewall farm FIRE1:

```
Router# show ip slb firewallfarm

firewall farm   hash      state       reals
---------------------------------------------
FIRE1           IPADDR    INSERVICE   2
```

How to Verify Firewall Connectivity

Perform the following optional task to verify firewall connectivity.

To verify that IOS SLB firewall load balancing is configured and is operating correctly, perform the following steps:

SUMMARY STEPS

1. Ping the external real servers (the ones outside the firewall) from the IOS SLB firewall load-balancing switch.
2. Ping the internal real servers (the ones inside the firewall) from the clients.
3. Use the **show ip slb stats** command to show information about the IOS SLB firewall load-balancing network status:
4. Use the **show ip slb real detail**command to show information about the IOS SLB firewall load-balancing real server status:
5. Use the **show ip slb conns**command to show information about the active IOS SLB firewall load-balancing connections:

DETAILED STEPS

Step 1   Ping the external real servers (the ones outside the firewall) from the IOS SLB firewall load-balancing switch.

Step 2   Ping the internal real servers (the ones inside the firewall) from the clients.

Step 3   Use the **show ip slb stats** command to show information about the IOS SLB firewall load-balancing network status:

Example:

```
Router# show ip slb stats
 Pkts via normal switching:  0
 Pkts via special switching: 0
 Pkts dropped:               0
 Connections Created:        1911871
 Connections Established:    1967754
 Connections Destroyed:      1313251
 Connections Reassigned:     0
 Zombie Count:               0
 Connections Reused:         59752
 Connection Flowcache Purges:1776582
 Failed Connection Allocs:   17945
 Failed Real Assignments:    0
```

- Normal switching exists when IOS SLB packets are managed on normal IOS switching paths (CEF, fast switching, and process level switching).
- Special switching exists when IOS SLB packets are managed on hardware-assisted switching paths.

Step 4   Use the **show ip slb real detail**command to show information about the IOS SLB firewall load-balancing real server status:

Example:

```
Router# show ip slb reals detail
172.16.88.5, SF1, state = OPERATIONAL, type = server
  ipv6 = 2342:2342:2343:FF04:2388:BB03:3223:8912
  conns = 0, dummy_conns = 0, maxconns = 4294967295
  weight = 8, weight(admin) = 8, metric = 0, remainder = 0
  reassign = 3, retry = 60
  failconn threshold = 8, failconn count = 0
  failclient threshold = 2, failclient count = 0
  total conns established = 0, total conn failures = 0
  server failures = 0
```

Step 5   Use the **show ip slb conns**command to show information about the active IOS SLB firewall load-balancing connections:

Example:

```
Router# show ip slb conns
vserver        prot client              real               state   nat
-----------------------------------------------------------------------------
FirewallTCP    TCP  80.80.50.187:40000   10.1.1.4           ESTAB   none
FirewallTCP    TCP  80.80.50.187:40000   10.1.1.4           ESTAB   none
FirewallTCP    TCP  80.80.50.187:40000   10.1.1.4           ESTAB   none
FirewallTCP    TCP  80.80.50.187:40000   10.1.1.4           ESTAB   none
FirewallTCP    TCP  80.80.50.187:40000   10.1.1.4           ESTAB   none
```

See "How to Monitor and Maintain the Cisco IOS SLB Feature" for additional commands used to verify IOS SLB networks and connections.

How to Configure a Probe

The following sections describe how to configure and verify probes. By default, no probes are configured in IOS SLB.

IOS SLB uses probes to verify connectivity and detect failures. For a detailed description of each type of probe, see the "Probes" section.

Perform the following task to configure a probe. Required and optional tasks are indicated.

How to Configure a Custom UDP Probe

Perform the following task to configure a custom User Datagram Protocol (UDP) probe.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb probe** *probe* **custom udp**
4. **address** [*ip-address*] [**routed**]
5. **faildetect** *number-of-probes*
6. **interval** *seconds*
7. **port** *port*
8. **request data** {*start-byte* | **continue**} *hex-data-string*
9. **response** *clause-number* **data** *start-byte hex-data-string*
10. **timeout** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal**	Enters global configuration mode.

Command or Action	Purpose
Example: `Router# configure terminal`	
Step 3 **ip slb probe** *probe* **custom udp** **Example:** `Router(config)# ip slb probe PROBE6 custom udp`	Configures the IOS SLB probe name and enters custom User Datagram Protocol (UDP) probe configuration mode.
Step 4 **address** [*ip-address*] [**routed**] **Example:** `Router(config-slb-probe)# address 10.1.1.1`	(Optional) Configures an IP address to which to send the custom UDP probe.
Step 5 **faildetect** *number-of-probes* **Example:** `Router(config-slb-probe)# faildetect 16`	(Optional) Specifies the number of consecutive unacknowledged custom UDP probes that constitute failure of the real server.
Step 6 **interval** *seconds* **Example:** `Router(config-slb-probe)# interval 11`	(Optional) Configures the custom UDP probe transmit timers.
Step 7 **port** *port* **Example:** `Router(config-slb-probe)# port 8`	Configures the port to which the custom UDP probe is to connect.
Step 8 **request data** {*start-byte* | **continue**} *hex-data-string* **Example:** `Router(config-slb-probe)# request data 0 05 04 00 77 18 2A D6 CD 0A AD 53 4D F1 29 29 CF C1 96 59 CB`	Defines the payload of the UDP request packet to be sent by a custom UDP probe.
Step 9 **response** *clause-number* **data** *start-byte hex-data-string* **Example:** `Router(config-slb-probe)# response 2 data 44 DD DD`	Defines the data string to match against custom UDP probe response packets.
Step 10 **timeout** *seconds* **Example:** `Router(config-slb-probe)# timeout 20`	(Optional) Sets a timeout for custom UDP probes.

How to Configure a DNS Probe

Perform the following task to configure a Domain Name System (DNS) probe.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb probe** *probe* **dns**
4. **address** [*ip-address* [**routed**]]
5. **faildetect** *number-of-probes*
6. **interval** *seconds*
7. **lookup** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**ip slb probe** *probe* **dns** **Example:** Router(config)# ip slb probe PROBE4 dns	Configures the IOS SLB probe name and enters Domain Name System (DNS) probe configuration mode.
Step 4	**address** [*ip-address* [**routed**]] **Example:** Router(config-slb-probe)# address 10.1.10.1	(Optional) Configures an IP address to which to send the DNS probe.
Step 5	**faildetect** *number-of-probes* **Example:** Router(config-slb-probe)# faildetect 16	(Optional) Specifies the number of consecutive unacknowledged DNS probes that constitute failure of the real server or firewall.
Step 6	**interval** *seconds* **Example:** Router(config-slb-probe)# interval 11	(Optional) Configures the DNS probe transmit timers.

Command or Action	Purpose
Step 7 **lookup** *ip-address* **Example:** `Router(config-slb-probe)# lookup 10.1.10.1`	(Optional) Configures an IP address of a real server that a DNS server should supply in response to a domain name resolve request.

How to Configure an HTTP Probe

Perform the following task to configure an HTTP probe.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb probe** *probe* **http**
4. **address** [*ip-address* [**routed**]]
5. **credentials** {*username* [*password*]}
6. **expect** [**status** *status-code*] [**regex** *expression*]
7. **header** *field-name* [*field-value*]
8. **interval** *seconds*
9. **port** *port*
10. **request** [**method** {**get** | **post** | **head** | **name** *name*}] [**url** *path*]
11. Configure a route to the virtual server.

DETAILED STEPS

Command or Action	Purpose
Step 1 **enable** **Example:** `Router> enable`	Enables privileged EXEC mode. If prompted, enter your password.
Step 2 **configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3 **ip slb probe** *probe* **http** **Example:** `Router(config)# ip slb probe PROBE2 http`	Configures the IOS SLB probe name and enters HTTP probe configuration mode.
Step 4 **address** [*ip-address* [**routed**]]	(Optional) Configures an IP address to which to send the HTTP probe.

	Command or Action	Purpose
	Example:	
	`Router(config-slb-probe)# address 10.1.10.1`	
Step 5	**credentials** {*username* [*password*]}	(Optional) Configures header values for the HTTP probe.
	Example:	
	`Router(config-slb-probe)# credentials Username1 password`	
Step 6	**expect** [**status** *status-code*] [**regex** *expression*]	(Optional) Configures the expected HTTP status code or regular expression.
	Example:	
	`Router(config-slb-probe)# expect status 401 regex Copyright`	
Step 7	**header** *field-name* [*field-value*]	(Optional) Configures header values for the HTTP probe.
	Example:	
	`Router(config-slb-probe)# header HeaderName HeaderValue`	
Step 8	**interval** *seconds*	(Optional) Configures the HTTP probe transmit timers.
	Example:	
	`Router(config-slb-probe)# interval 11`	
Step 9	**port** *port*	(Optional) Configures the port to which the HTTP probe is to connect.
	Example:	
	`Router(config-slb-probe)# port 8`	
Step 10	**request** [**method** {**get** | **post** | **head** | **name** *name*}] [**url** *path*]	(Optional) Configures the URL path to request from the server, and the method used to perform the request to the server.
	Example:	
	`Router(config-slb-probe)# request method post url /probe.cgi?all`	
Step 11	Configure a route to the virtual server.	HTTP probes require a route to the virtual server. The route is not used, but it must exist to enable the socket code to verify that the destination can be reached, which in turn is essential for HTTP probes to function correctly. The route can be either: • Host route--Advertised by the virtual server • Default route--Specified using the **ip route 0.0.0.0 0.0.0.0**command, for example

How to Configure a Ping Probe

Perform the following task to configure a ping probe.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb probe** *probe* **ping**
4. **address** [*ip-address* [**routed**]]
5. **faildetect** *number-of-pings*
6. **interval** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**ip slb probe** *probe* **ping** **Example:** Router(config)# ip slb probe PROBE1 ping	Configures the IOS SLB probe name and enters ping probe configuration mode.
Step 4	**address** [*ip-address* [**routed**]] **Example:** Router(config-slb-probe)# address 10.1.10.1	(Optional) Configures an IP address to which to send the ping probe.
Step 5	**faildetect** *number-of-pings* **Example:** Router(config-slb-probe)# faildetect 16	(Optional) Specifies the number of consecutive unacknowledged pings that constitute failure of the real server or firewall.
Step 6	**interval** *seconds* **Example:** Router(config-slb-probe)# interval 11	(Optional) Configures the ping probe transmit timers.

How to Configure a TCP Probe

Perform the following task to configure a TCP probe.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb probe** *probe* **tcp**
4. **address** [*ip-address* [**routed**]]
5. **interval** *seconds*
6. **port** *port*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**ip slb probe** *probe* **tcp** **Example:** Router(config)# ip slb probe PROBE5 tcp	Configures the IOS SLB probe name and enters TCP probe configuration mode.
Step 4	**address** [*ip-address* [**routed**]] **Example:** Router(config-slb-probe)# address 10.1.10.1	(Optional) Configures an IP address to which to send the TCP probe.
Step 5	**interval** *seconds* **Example:** Router(config-slb-probe)# interval 5	(Optional) Configures the TCP probe transmit timers.
Step 6	**port** *port* **Example:** Router(config-slb-probe)# port 8	Configures the port to which the TCP probe is to connect.

How to Configure a WSP Probe

Perform the following task to configure a Wireless Session Protocol (WSP) probe.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb probe** *probe* **wsp**
4. **address** [*ip-address* [**routed**]]
5. **interval** *seconds*
6. **url** [*path*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**ip slb probe** *probe* **wsp** **Example:** Router(config)# ip slb probe PROBE3 wsp	Configures the IOS SLB probe name and enters Wireless Session Protocol (WSP) probe configuration mode.
Step 4	**address** [*ip-address* [**routed**]] **Example:** Router(config-slb-probe)# address 10.1.10.1	(Optional) Configures an IP address to which to send the WSP probe.
Step 5	**interval** *seconds* **Example:** Router(config-slb-probe)# interval 11	(Optional) Configures the WSP probe transmit timers.
Step 6	**url** [*path*] **Example:** Router(config-slb-probe)# url http://localhost/test.txt	(Optional) Configures the WSP probe URL path.

How to Associate a Probe

Perform the following task to associate a probe with a real server or firewall.

After configuring a probe, you must associate the probe with a real server or firewall using the **probe** command. See "How to Configure a Server Farm and a Real Server" and "How to Configure Firewall Load Balancing" for more details.

> ✎
>
> **Note**   You cannot associate a WSP probe with a firewall.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **ip slb firewallfarm** *firewall-farm*
   -
   - **ip slb serverfarm** *server-farm*
4. Do one of the following:
   - **probe** *probe*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	Do one of the following: • **ip slb firewallfarm** *firewall-farm* • • **ip slb serverfarm** *server-farm* **Example:** Router(config)# ip slb serverfarm PUBLIC	Identifies a firewall farm and enters firewall farm configuration mode. or Identifies a server farm and enters SLB server farm configuration mode.

Command or Action	Purpose
Example: `Router(config)# ip slb firewallfarm FIRE1`	

Step 4 Do one of the following:
 	
• **probe** *probe*
 	
Example:
 	
`Router(config-slb-sfarm)# probe PROBE1`
 	
Example:
 	
`Router(config-slb-fw-real)# probe FireProbe`
	Associates a probe with a firewall farm or a server farm.

How to Verify a Probe

Perform the following optional task to verify a probe.

To verify that a probe is configured correctly, use the **show ip slb probe** command:

```
Router# show ip slb probe
Server:Port          State          Outages  Current  Cumulative
---------------------------------------------------------------
10.1.1.1:80          OPERATIONAL         0  never    00:00:00
10.1.1.2:80          OPERATIONAL         0  never    00:00:00
10.1.1.3:80          OPERATIONAL         0  never    00:00:00
```

How to Configure DFP

Perform the following task to configure IOS SLB as a Dynamic Feedback Protocol (DFP) manager, and to identify a DFP agent with which IOS SLB can initiate connections.

You can define IOS SLB as a DFP manager, as a DFP agent for another DFP manager, 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb dfp** [**password** [[*encrypt*] *secret-string* [*timeout*]]
4. **agent** *ip-address port* [*timeout* [*retry-count* [*retry-interval*]]]
5. Configure IOS SLB as a DFP agent.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**ip slb dfp** [**password**[[*encrypt*] *secret-string* [*timeout*]] **Example:** Router(config)# ip slb dfp password Password1 360	Configures Dynamic Feedback Protocol (DFP), supplies an optional password, and enters DFP configuration mode.
Step 4	**agent** *ip-address port* [*timeout*[*retry-count* [*retry-interval*]]] **Example:** Router(config-slb-dfp)# agent 10.1.1.1 2221 30 0 10	Identifies a DFP agent to which IOS SLB can connect.
Step 5	Configure IOS SLB as a DFP agent.	To configure IOS SLB as a DFP agent, refer to the *DFP Agent Subsystem* feature document for Cisco IOS Release 12.2(18)SXB.

GPRS Load Balancing Configuration Task List

Perform the following tasks to configure general packet radio service (GPRS) load balancing.

SUMMARY STEPS

1. Configure a server farm and a real server.
2. Configure a virtual server.
3. Configure the virtual IP address as a loopback on each of the GGSNs in the servers.
4. Route each GGSN to each associated SGSN.
5. Route each SGSN to the virtual templates on each associated Cisco GGSN, and to the GPRS load-balancing virtual server.
6. Configure a GSN idle timer.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Configure a server farm and a real server.	See "How to Configure a Server Farm and a Real Server".

	Command or Action	Purpose

		When you configure the server farm and real server for GPRS load balancing, keep the following considerations in mind:
		
		 • If GTP cause code inspection:
		    ◦ Is not enabled--Accept the default setting (the weighted round robin algorithm) for the **predictor** command.
		    ◦ Is enabled--Specify either the weighted round robin (**roundrobin**) or the weighted least connections (**leastconns**) algorithm.
		 • Specify the IP addresses (virtual template addresses for Cisco GGSNs) of the real servers performing the GGSN function, using the **real** command.
		 • Specify a reassign threshold less than the SGSN's N3-REQUESTS counter value using the **reassign** command.
		 • To enable dual-stack support for GTP load balancing:
		    ◦ Specify the real server's IPv6 address using the **real**command.

Step 2	Configure a virtual server.	See "How to Configure a Virtual Server".
		
		 When you configure the **virtual** command, keep the following considerations in mind:
		
		 • Specify a virtual GGSN IP address as the virtual server, and specify the **udp** keyword option.
		 • To load-balance GTP v1 and GTP v2 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 GPRS load balancing:
		    ◦ *Without* GTP cause code inspection--Specify the **service gtp**keyword option.
		
		 In GPRS 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.
		
		 • ◦ *With* GTP cause code inspection--Specify the **service gtp-inspect**keyword option.
		 • To enable dual-stack support for GTP load balancing:
		    ◦ Specify the virtual server's IPv6 address and optional IPv6 prefix, using the **virtual** command.
		    ◦ Associate the primary IPv6 server farm and optional backup IPv6 server farm with the virtual server, using the **serverfarm**command.
		    ◦ Remove the **client** command from the configuration.

Step 3	Configure the virtual IP address as a loopback on each of the GGSNs in the servers.	(Required for dispatched mode) This step is required only if you are using dispatched mode *without* GTP cause code inspection enabled. Refer to the *Cisco IOS Interface Configuration Guide* "Configuring Virtual Interfaces" section for more information.

	Command or Action	Purpose
Step 4	Route each GGSN to each associated SGSN.	The route can be static or dynamic, but the GGSN needs to be able to reach the SGSN. Refer to the *Cisco IOS Mobile Wireless Configuration Guide* "Configuring Network Access to the GGSN" section for more details.
Step 5	Route each SGSN to the virtual templates on each associated Cisco GGSN, and to the GPRS load-balancing virtual server.	(Required) Refer to the configuration guide for your SGSN for more details.
Step 6	Configure a GSN idle timer.	(Optional) This step is applicable only if GTP cause code inspection is enabled. See "How to Configure a GSN Idle Timer" for more information.

How to Configure a GSN Idle Timer

Perform this task to configure a GPRS support node (GSN) idle timer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb timers gtp gsn** *duration*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip slb timers gtp gsn** *duration* **Example:** `Router(config)# ip slb timers gtp gsn 45`	Change the amount of time IOS SLB maintains sessions to and from an idle gateway GPRS support node (GGSN) or serving GPRS support node (SGSN).

GGSN-IOS SLB Messaging Task List

Perform this task to configure GGSN-IOS SLB messaging.

SUMMARY STEPS

1. Configure the GGSN to support GGSN-IOS SLB messaging.
2. Configure a server farm and a real server.
3. Configure a virtual server.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Configure the GGSN to support GGSN-IOS SLB messaging.	When you configure GGSN-IOS SLB messaging support, configure all IOS SLB virtual servers that share the same GGSN to use the same NAT mode, either dispatched mode or directed mode, using the **gprs slb mode**command. The virtual servers cannot use a mix of dispatched mode and directed mode, because you can configure only one NAT mode on a given GGSN.
		For more information, refer to the *Cisco IOS Mobile Wireless Configuration Guide* for GGSN Release 5.0 for Cisco IOS Release 12.3(2)XU or later.
Step 2	Configure a server farm and a real server.	See "How to Configure a Server Farm and a Real Server".
		When you configure the server farm and real server for GGSN-IOS SLB messaging, to prevent IOS SLB from failing the current real server when reassigning the session to a new real server, disable automatic server failure detection by specifying the **no faildetect inband** command.
Step 3	Configure a virtual server.	See "How to Configure a Virtual Server".
		When you configure the virtual server for GGSN-IOS SLB messaging, specify the **gtp notification cac**command to limit the number of times IOS SLB can reassign a session to a new real server.

How to Configure GPRS Load Balancing Maps

Perform this task to configure GPRS load balancing maps.

GPRS load balancing maps enable IOS SLB to categorize and route user traffic based on access point names (APNs). To enable maps for GPRS load balancing, you must define a GPRS Tunneling Protocol (GTP) map, then associate the map with a server farm.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb map** *map-id* **gtp** | **radius**}
4. **apn** *string*
5. **exit**
6. **ip slb vserver** *virtual-server*
7. **virtual** *ipv4-address* [*ipv4-netmask*[**group**]] [**ipv6** *ipv6-address* [**prefix** *ipv6-prefix*]] {**tcp**| **udp**} [*port*| **any**] [**service** *service*]
8. **serverfarm** *primary-farm* [**backup** *backup-farm*[**sticky**]] [**ipv6-primary** *ipv6-primary-farm*[**ipv6-backup** *ipv6-backup-farm*]] [**map** *map-id* **priority** *priority*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**ip slb map** *map-id* **gtp** | **radius**} **Example:** Router(config)# ip slb map 1 radius	Configures an IOS SLB GTP map and enters SLB GTP map configuration mode.
Step 4	**apn** *string* **Example:** Router(config-slb-map-gtp)# apn abc	Configures an ASCII regular expression string to be matched against the access point name (APN) for general packet radio service (GPRS) load balancing.
Step 5	**exit** **Example:** Router(config-slb-map-gtp)# exit	Exits SLB GTP map configuration mode.
Step 6	**ip slb vserver** *virtual-server* **Example:** Router(config)# ip slb vserver GGSN_SERVER	Identifies a virtual server and enters virtual server configuration mode.
Step 7	**virtual** *ipv4-address* [*ipv4-netmask*[**group**]] [**ipv6** *ipv6-address* [**prefix** *ipv6-prefix*]] {**tcp**| **udp**} [*port*| **any**] [**service** *service*] **Example:** Router(config-slb-vserver)# virtual 10.10.10.10 udp 0 service gtp	Specifies the virtual server IP address, type of connection, and optional TCP or User Datagram Protocol (UDP) port number, Internet Key Exchange (IKE) or Wireless Session Protocol (WSP) setting, and service coupling. **Note**For GPRS load balancing: • ◦ Specify a virtual GGSN IP address as the virtual server, and specify the **udp** keyword option. ◦ To load-balance GTP v1 and GTP v2 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).

Command or Action	Purpose
	◦ 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 GPRS load balancing *without* GTP cause code inspection, specify the **service gtp** keyword option. ◦ To enable GPRS load balancing *with* GTP cause code inspection, specify the **service gtp-inspect** keyword option. ◦ For dual-stack support for GTP load balancing, specify the virtual server's IPv4 and IPv6 addresses and optional IPv6 prefix.
Step 8 **serverfarm** *primary-farm* [**backup** *backup-farm*[**sticky**]] [**ipv6-primary** *ipv6-primary-farm*[**ipv6-backup** *ipv6-backup-farm*]] [**map** *map-id* **priority** *priority*] **Example:** `Router(config-slb-vserver)# serverfarm farm1 backup farm2 map 1 priority 3`	Associates a GTP map with a server farm. Associates a real server farm with a virtual server, and optionally configures a backup server farm and specifies that sticky connections are to be used in the backup server farm. **Note** For GPRS load balancing, if a real server is defined in two or more server farms, each server farm must be associated with a different virtual server. You can associate more than one server farm with a virtual server by configuring more than one **serverfarm** command, each with a unique map ID and a unique priority. (That is, each map ID and each map priority must be unique across all server farms associated with the virtual server.) If you are using GTP maps, and you have configured a real server in more than one server farm, you must associate a different virtual server with each server farm.

How to Configure KAL-AP Agent Support

Perform this task to configure KeepAlive Application Protocol (KAL-AP) agent support.

KAL-AP agent support enables IOS SLB to perform load balancing in a global server load balancing (GSLB) environment.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb capp udp**
4. **peer** [*ip-address*] **port** *port*
5. **peer** [*ip-address*] **secret** [*encrypt*] *secret-string*
6. **exit**
7. **ip slb serverfarm** *server-farm*
8. **kal-ap domain** *tag*
9. **farm-weight** *setting*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**ip slb capp udp** **Example:** Router(config)# ip slb capp udp	Enables the KAL-AP agent and enters SLB Content Application Peering Protocol (CAPP) configuration mode.
Step 4	**peer** [*ip-address*] **port** *port* **Example:** Router(config-slb-capp)# peer port 6000	(Optional) Specifies the port to which the KAL-AP agent is to connect.
Step 5	**peer** [*ip-address*] **secret** [*encrypt*] *secret-string* **Example:** Router(config-slb-capp)# peer secret SECRET_STRING	(Optional) Enables Message Digest Algorithm Version 5 (MD5) authentication for the KAL-AP agent.
Step 6	**exit** **Example:** Router(config-slb-map-gtp)# exit	Exits SLB CAPP configuration mode.
Step 7	**ip slb serverfarm** *server-farm* **Example:** Router(config)# ip slb serverfarm PUBLIC	Identifies a server farm and enters SLB server farm configuration mode.
Step 8	**kal-ap domain** *tag* **Example:** Router(config-slb-sfarm)# kal-ap domain chicago-com	(Optional) Enables the KAL-AP agent to look for a domain tag when reporting the load for a virtual server.
Step 9	**farm-weight** *setting*	(Optional) Specifies a weight to be used by the KAL-AP agent when calculating the load value for a server farm.

Command or Action	Purpose
Example: `Router(config-slb-sfarm)# farm-weight 16`	

RADIUS Load Balancing Configuration Task List

Perform this task to configure RADIUS load balancing.

SUMMARY STEPS

1. Configure a server farm and a real server.
2. Configure a virtual server.
3. Configure a virtual server. (continued)
4. Enable IOS SLB to inspect packets for RADIUS framed-IP sticky routing.
5. Configure RADIUS load balancing maps.
6. Configure RADIUS load balancing accelerated data plane forwarding.
7. Increase the number of available MLS entries.
8. Configure a probe.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Configure a server farm and a real server.	See "How to Configure a Server Farm and a Real Server". When you configure the server farm and real server for RADIUS load balancing, keep the following considerations in mind: • Accept the default setting (the weighted round robin algorithm) for the **predictor** command. • (Optional) To enable session-based failure detection, specify a value of 1 for the **numclients** keyword on the **faildetect numconns**command. • (Optional) To specify the maximum number of IOS SLB RADIUS and GTP sticky subscribers that can be assigned to an individual virtual server, use the **maxclients** command.
Step 2	Configure a virtual server.	See "How to Configure a Virtual Server". When you configure the virtual server for RADIUS load balancing, keep the following considerations in mind: • Specify the **service radius**keyword option, using the **virtual** command. • (Optional) To enable framed-IP routing to inspect the ingress interface, specify the **access**_interface_ **route framed-ip**command. If you configure the **access**_interface_ **route framed-ip**command, you must also configure the **virtual** command with the **service radius**keywords specified. • (Optional) To change the amount of time IOS SLB waits for an ACCT-START message from a new mobile IP foreign agent in the event of a foreign agent hand-off, configure a **hand-off radius**command.

	Command or Action	Purpose
		• (Optional) To set a duration for RADIUS entries in the IOS SLB session database, configure an **idle** command with the **radius request** keywords specified. • (Optional) To set a duration for entries in the IOS SLB RADIUS framed-IP sticky database, configure an **idle** command with the **radius framed-ip** keywords specified.
Step 3	Configure a virtual server. (continued)	• (Optional) To enable IOS SLB to create the IOS SLB RADIUS framed-IP sticky database and direct RADIUS requests and non-RADIUS flows from a subscriber to the same service gateway, specify the **sticky** command with the **radius framed-ip** keywords. If you configure the **sticky radius framed-ip** command, you must also configure the **virtual** command with the **service radius** keywords specified. • (Optional) To enable IOS SLB to purge entries in the IOS SLB RADIUS framed-IP sticky database upon receipt of an Accounting On or Off message, specify the **purge radius framed-ip acct on-off virtual server** configuration command. To prevent IOS SLB from purging entries in the IOS SLB RADIUS framed-IP sticky database upon receipt of an Accounting On or Off message, specify the **no purge radius framed-ip acct on-off virtual server** configuration command. • (Optional) To enable IOS SLB to purge entries in the IOS SLB RADIUS framed-IP sticky database upon receipt of an Accounting-Stop message, specify the **purge radius framed-ip acct stop virtual server** configuration command. To prevent IOS SLB from purging entries in the IOS SLB RADIUS framed-IP sticky database upon receipt of an Accounting-Stop message, specify the **no purge radius framed-ip acct stop virtual server** configuration command. • (Optional--For CDMA2000 networks only) To enable IOS SLB to create the IOS SLB RADIUS calling-station-ID sticky database and direct RADIUS requests from a subscriber to the same service gateway based on the calling station ID, specify the **sticky** command with the **radius calling-station-id** keywords. To enable IOS SLB to create the IOS SLB RADIUS username sticky database and direct RADIUS requests from a subscriber to the same service gateway based on the username, specify the **sticky** command with the **radius username** keywords. If you configure the **sticky radius calling-station-id** command or the **sticky radius username** command, you must also configure the **virtual** command with the **service radius** keywords specified, and you must configure the **sticky radius framed-ip** command. You cannot configure both the **sticky radius calling-station-id** command and the **sticky radius username** command on the same virtual server. • (Optional--For RADIUS load balancing accelerated data plane forwarding only) To configure a VSA correlation group for an authentication virtual server, and to specify whether IOS SLB is to create VSA correlation entries based on RADIUS calling station IDs or RADIUS usernames, configure the **radius inject auth** command. To configure a timer for VSA correlation for an authentication virtual server, configure the **radius inject auth timer** command. To buffer VSAs for VSA correlation for an authentication virtual server, configure the **radius inject auth vsa** command.

	Command or Action	Purpose
		To configure a VSA correlation group for an accounting virtual server, and to enable Message Digest Algorithm Version 5 (MD5) authentication for VSA correlation, configure the **radius inject acct** command.
Step 4	Enable IOS SLB to inspect packets for RADIUS framed-IP sticky routing.	(Optional) See "How to Enable IOS SLB to Inspect Packets for RADIUS Framed-IP Sticky Routing".
Step 5	Configure RADIUS load balancing maps.	(Optional) See "How to Configure RADIUS Load Balancing Maps".
Step 6	Configure RADIUS load balancing accelerated data plane forwarding.	(Optional) See "How to Configure RADIUS Load Balancing Accelerated Data Plane Forwarding".
Step 7	Increase the number of available MLS entries.	(Optional) If you are running IOS SLB in dispatched mode on a Cisco Catalyst 6500 series switch with Cisco Supervisor Engine 2, you can improve performance by configuring the **no mls netflow** command. This command increases the number of MLS entries available for hardware switching of end-user flows. **Note**If you are using IOS features that use the hardware NetFlow table, such as microflow QoS, reflexive ACLs, TCP intercept, or Web Cache Redirect, do not configure the **no mls netflow** command. For more information about configuring MLS NetFlow, refer to the *Cisco Catalyst 6000 Family IOS Software Configuration Guide* .
Step 8	Configure a probe.	See "How to Configure a Probe". To verify the health of the server, configure a ping probe.

How to Enable IOS SLB to Inspect Packets for RADIUS Framed-IP Sticky Routing

You can enable IOS SLB to inspect packets whose source IP addresses match a configured IP address and subnet mask. If the source IP address of an inspected packet matches an entry in the IOS SLB RADIUS framed-IP sticky database, IOS SLB uses that entry to route the packet. Otherwise, IOS routes the packet.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb route framed-ip deny** | *ip-address netmask* **framed-ip** | **inter-firewall**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**ip slb route framed-ip deny** | *ip-address netmask* **framed-ip** | **inter-firewall** **Example:** Router(config)# ip slb route 10.10.10.1 255.255.255.255 framed-ip	Enables IOS SLB to route packets using the RADIUS framed-IP sticky database, or to route packets from one firewall real server back through another firewall real server.

How to Configure RADIUS Load Balancing Maps

RADIUS load balancing maps enable IOS SLB to categorize and route user traffic based on RADIUS calling station IDs and usernames. To enable maps for RADIUS load balancing, you must define a RADIUS map, then associate the map with a server farm.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb map** *map-id* **radius**
4. **calling-station-id** *string*
5. **username** *string*
6. **exit**
7. **ip slb vserver** *virtual-server*
8. **virtual** *ipv4-address* [*ipv4-netmask*[**group**]] [**ipv6** *ipv6-address* [**prefix** *ipv6-prefix*]] {**tcp**\| **udp**} [*port*\| **any**] [**service** *service*]
9. **serverfarm** *primary-farm* [**backup** *backup-farm*[**sticky**]] [**ipv6-primary** *ipv6-primary-farm*[**ipv6-backup** *ipv6-backup-farm*]] [**map** *map-id* **priority** *priority*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. If prompted, enter your password.

Command or Action	Purpose
Example: `Router> enable`	
Step 2 **configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3 **ip slb map** *map-id* **radius** **Example:** `Router(config)# ip slb map 1 radius`	Configures an IOS SLB RADIUS map and enters SLB RADIUS map configuration mode.
Step 4 **calling-station-id** *string* **Example:** `Router(config-slb-radius-map)# calling-station-id .919*`	Configures an ASCII regular expression string to be matched against the calling station ID attribute for RADIUS load balancing.
Step 5 **username** *string* **Example:** `Router(config-slb-map-radius)# )# username ...?525*`	Configures an ASCII regular expression string to be matched against the username attribute for RADIUS load balancing.
Step 6 **exit** **Example:** `Router(config-slb-map-gtp)# exit`	Exits SLB RADIUS map configuration mode.
Step 7 **ip slb vserver** *virtual-server* **Example:** `Router(config)# ip slb vserver GGSN_SERVER`	Identifies a virtual server and enters virtual server configuration mode.
Step 8 **virtual** *ipv4-address* [*ipv4-netmask*[**group**]] [**ipv6** *ipv6-address* [**prefix** *ipv6-prefix*]] {**tcp**| **udp**} [*port*| **any**] [**service** *service*] **Example:** `Router(config-slb-vserver)# virtual 10.0.0.1 udp 0 service radius`	Specifies the virtual server IP address, type of connection, and optional TCP or User Datagram Protocol (UDP) port number, Internet Key Exchange (IKE) or Wireless Session Protocol (WSP) setting, and service coupling. **Note** For RADIUS load balancing, specify the **service radius** keyword option.
Step 9 **serverfarm** *primary-farm* [**backup** *backup-farm*[**sticky**]] [**ipv6-primary** *ipv6-primary-*	Associates a RADIUS map with a server farm. Associates a real server farm with a virtual server, and optionally configures a backup server

Command or Action	Purpose
farm[**ipv6-backup** *ipv6-backup-farm*]] [**map** *map-id* **priority** *priority*] **Example:** `Router(config-slb-vserver)# serverfarm SF1 backup SF2 map 1 priority 1`	farm and specifies that sticky connections are to be used in the backup server farm. **Note**RADIUS load balancing does not support the **sticky**keyword. You can associate more than one server farm with a virtual server by configuring more than one **serverfarm** command, each with a unique map ID and a unique priority. (That is, each map ID and each map priority must be unique across all server farms associated with the virtual server.)

How to Configure RADIUS Load Balancing Accelerated Data Plane Forwarding

Perform this task to configure RADIUS load balancing accelerated data plane forwarding.

RADIUS load balancing accelerated data plane forwarding, also known as Turbo RADIUS load balancing, is a high-performance solution that uses basic policy-based routing (PBR) route maps to manage subscriber data-plane traffic in a Cisco Content Services Gateway (CSG) environment.

Turbo RADIUS load balancing requires a server farm configured with **predictor route-map** on the accounting virtual server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb serverfarm** *server-farm*
4. **predictor** [**roundrobin**| **leastconns**| **route-map** *mapname*]
5. **exit**
6. **ip slb vserver** *virtual-server*
7. **virtual** *ipv4-address* [*ipv4-netmask*[**group**]] [**ipv6** *ipv6-address* [**prefix** *ipv6-prefix*]] {**tcp**| **udp**} [*port*| **any**] [**service** *service*]
8. **serverfarm** *primary-farm* [**backup** *backup-farm*[**sticky**]] [**ipv6-primary** *ipv6-primary-farm*[**ipv6-backup** *ipv6-backup-farm*]] [**map** *map-id* **priority** *priority*]
9. **radius acct local-ack key** [*encrypt*] *secret-string*
10. **radius inject auth** *group-number* {**calling-station-id**| **username**}
11. **radius inject auth timer** *seconds*
12. **radius inject auth vsa** *vendor-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. If prompted, enter your password.

	Command or Action	Purpose
	Example: Router> enable	
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**ip slb serverfarm** *server-farm* **Example:** Router(config)# ip slb serverfarm PUBLIC	Identifies a server farm and enters SLB server farm configuration mode.
Step 4	**predictor** [**roundrobin**| **leastconns**| **route-map** *mapname*] **Example:** Router(config-slb-sfarm)# predictor route-map map1	(Optional) Specifies the algorithm to be used to determine how a real server is selected. Turbo RADIUS load balancing requires the **route-map** keyword and *mapname* argument. When you specify the **predictor route-map** command, no further commands in SLB server farm configuration mode or real server configuration mode are allowed.
Step 5	**exit** **Example:** Router(config-slb-sfarm)# exit	Exits SLB server farm configuration mode.
Step 6	**ip slb vserver** *virtual-server* **Example:** Router(config)# ip slb vserver RADIUS_AUTH	Identifies a virtual server and enters virtual server configuration mode.
Step 7	**virtual** *ipv4-address* [*ipv4-netmask*[**group**]] [**ipv6** *ipv6-address* [**prefix** *ipv6-prefix*]] {**tcp**| **udp**} [*port*| **any**] [**service** *service*] **Example:** Router(config-slb-vserver)# virtual 10.10.10.10 udp 1813 service radius	Specifies the virtual server IP address, type of connection, and optional TCP or User Datagram Protocol (UDP) port number, Internet Key Exchange (IKE) or Wireless Session Protocol (WSP) setting, and service coupling and enters SLB virtual server configuration mode. **Note**For RADIUS load balancing, specify the **service radius**keyword option.
Step 8	**serverfarm** *primary-farm* [**backup** *backup-farm*[**sticky**]] [**ipv6-primary** *ipv6-primary-farm*[**ipv6-backup** *ipv6-backup-farm*]] [**map** *map-id* **priority** *priority*]	Associates a RADIUS map with a server farm. Associates a real server farm with a virtual server, and optionally configures a backup server farm and specifies that sticky connections are to be used in the backup server farm. **Note**RADIUS load balancing does not support the **sticky**keyword.

	Command or Action	Purpose
	Example: `Router(config-slb-vserver)# serverfarm AAAFARM`	You can associate more than one server farm with a virtual server by configuring more than one **serverfarm** command, each with a unique map ID and a unique priority. (That is, each map ID and each map priority must be unique across all server farms associated with the virtual server.)
Step 9	**radius acct local-ack key** [*encrypt*] *secret-string* **Example:** `Router(config-slb-vserver)# radius acct local-ack key SECRET_PASSWORD`	(Optional) Configures VSA correlation and enables a RADIUS virtual server to acknowledge RADIUS accounting messages **Note** If vendor-specific attribute (VSA) correlation is configured, and if the Cisco VSA is buffered, then the Cisco VSA is injected into the RADIUS Accounting-Start packet. Turbo RADIUS load balancing does not require VSA correlation. This command is valid only for VSA correlation accounting virtual servers.
Step 10	**radius inject auth** *group-number* {**calling-station-id**| **username**} **Example:** `Router(config-slb-vserver)# radius inject auth 1 calling-station-id`	(Optional) Configures a VSA correlation group for an IOS SLB RADIUS load balancing accelerated data plane forwarding authentication virtual server, and specifies whether IOS SLB is to create VSA correlation entries based on RADIUS calling station IDs or RADIUS usernames. For a given authentication virtual server, you can configure one **radius inject auth***group-number* **calling-station-id** command or one **radius inject auth***group-number* **username**command, but not both. This command is valid only for VSA correlation authentication virtual servers.
Step 11	**radius inject auth timer** *seconds* **Example:** `Router(config-slb-vserver)# radius inject auth timer 45`	(Optional) Configures a timer for VSA correlation for an IOS SLB RADIUS load balancing accelerated data plane forwarding authentication virtual server. This command is valid only for VSA correlation authentication virtual servers.
Step 12	**radius inject auth vsa** *vendor-id* **Example:** `Router(config-slb-vserver)# radius inject auth vsa vendor1`	(Optional) Buffers VSAs for VSA correlation for an IOS SLB RADIUS load balancing accelerated data plane forwarding authentication virtual server. This command is valid only for VSA correlation authentication virtual servers.

Exchange Director for mSEF Configuration Task List

Perform this task to configure Exchange Director for mobile Service Exchange Framework (mSEF).

-
-

RADIUS Configuration for the Exchange Director

Perform this task to configure RADIUS load balancing for the Exchange Director.

SUMMARY STEPS

1. Configure a server farm and a real server.
2. Configure a virtual server.
3. Configure a virtual server. (continued)
4. Enable IOS SLB to inspect packets for RADIUS framed-IP sticky routing.
5. Configure RADIUS load balancing maps.
6. Increase the number of available MLS entries.
7. Configure a probe.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Configure a server farm and a real server.	See "How to Configure a Server Farm and a Real Server".
		When you configure the server farm and real server for RADIUS for the Exchange Director, keep the following considerations in mind:
		• (Optional) Specify a value of 1 for the **numclients** keyword on the **faildetect numconns** command, if you want to enable session-based failure detection.
		• (Optional) To specify the maximum number of IOS SLB RADIUS and GTP sticky subscribers that can be assigned to an individual virtual server, use the **maxclients** command.
Step 2	Configure a virtual server.	See "How to Configure a Virtual Server".
		When you configure the virtual server for RADIUS for the Exchange Director, keep the following considerations in mind:
		• Specify the **service radius** keyword option, using the **virtual** command.
		• (Optional) To enable framed-IP routing to inspect the ingress interface, specify the **access** *interface* **route framed-ip** command.
		If you configure the **access** *interface* **route framed-ip** command, you must also configure the **virtual** command with the **service radius** keywords specified.
		• (Optional) To change the amount of time IOS SLB waits for an ACCT-START message from a new Mobile IP foreign agent in the event of a foreign agent hand-off, configure a **hand-off radius** command.
		• (Optional) To set a duration for RADIUS entries in the IOS SLB session database, configure an **idle** command with the **radius request** keywords specified.
		• (Optional) To set a duration for entries in the IOS SLB RADIUS framed-IP sticky database, configure an **idle** command with the **radius framed-ip** keywords specified.
		• (Optional) To enable IOS SLB to create the IOS SLB RADIUS framed-IP sticky database and direct RADIUS requests and non-RADIUS flows from a subscriber to the same service gateway, specify the **sticky** command with the **radius framed-ip** keywords.
		If you configure the **sticky radius framed-ip** command, you must also configure the **virtual** command with the **service radius** keywords specified.
Step 3	Configure a virtual server. (continued)	• (Optional--for CDMA2000 networks only) To enable IOS SLB to create the IOS SLB RADIUS calling-station-ID sticky database and direct RADIUS requests from a subscriber to the same service gateway based on the calling station ID, specify the **sticky** command with the **radius calling-station-id** keywords.

	Command or Action	Purpose
		To enable IOS SLB to create the IOS SLB RADIUS username sticky database and direct RADIUS requests from a subscriber to the same service gateway based on the username, specify the **sticky**command with the **radius username**keywords.
		If you configure the **sticky radius calling-station-id**command or the **sticky radius username**command, you must also configure the **virtual** command with the **service radius**keywords specified, and you must configure the **sticky radius framed-ip**command.
		You cannot configure both the **sticky radius calling-station-id** command and the **sticky radius username** command on the same virtual server.
Step 4	Enable IOS SLB to inspect packets for RADIUS framed-IP sticky routing.	(Optional) See "How to Enable IOS SLB to Inspect Packets for RADIUS Framed-IP Sticky Routing".
Step 5	Configure RADIUS load balancing maps.	(Optional) See "How to Configure RADIUS Load Balancing Maps".
Step 6	Increase the number of available MLS entries.	(Optional
Step 7	Configure a probe.	See "How to Configure a Probe". To verify the health of the server, configure a ping probe.

Firewall Configuration for the Exchange Director

Perform this task to configure firewall load balancing for the Exchange Director.

This section lists the tasks used to configure firewalls for the Exchange Director. Detailed configuration information is contained in the referenced sections of this or other documents. Required and optional tasks are indicated.

How to Configure a Firewall Farm

Perform the following required task to configure a firewall farm.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb firewallfarm** *firewall-farm*
4. **real** *ip-address*
5. **probe** *probe*
6. **weight** *setting*
7. **inservice**
8. **exit**
9. **access** [**source** *source-ip netmask*] [**destination** *destination-ip netmask*]| **inbound** *inbound-interface* | **outbound** *outbound-interface*]
10. **predictor hash address** [**port**]
11. **purge connection**
12. **purge sticky**
13. **replicate casa** *listen-ip remote-ip port* [*interval*] [**password** [[*encrypt*] *secret-string* [*timeout*]]]
14. **protocol tcp**
15. **delay** *duration*
16. **idle** *duration*
17. **maxconns** *maximum-number*
18. **sticky** *seconds* [**netmask** *netmask*] [**source**| **destination**]
19. **exit**
20. **protocol datagram**
21. **idle** *duration*
22. **maxconns** *maximum-number*
23. **sticky** *seconds* [**netmask** *netmask*] [**source**| **destination**]
24. **exit**
25. **inservice**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**ip slb firewallfarm** *firewall-farm*	Adds a firewall farm definition to the IOS SLB configuration and enters firewall farm configuration mode.

	Command or Action	Purpose
	Example: `Router(config)# ip slb firewallfarm FIRE1`	
Step 4	**real** *ip-address* **Example:** `Router(config-slb-fw)# real 10.1.1.1`	Identifies a firewall by IP address as a member of a firewall farm and enters real server configuration mode.
Step 5	**probe** *probe* **Example:** `Router(config-slb-fw-real)# probe FireProbe`	Associates a probe with the firewall.
Step 6	**weight** *setting* **Example:** `Router(config-slb-fw-real)# weight 16`	(Optional) Specifies the firewall's workload capacity relative to other firewalls in the firewall farm.
Step 7	**inservice** **Example:** `Router(config-slb-fw-real)# inservice`	Enables the firewall for use by the firewall farm and by IOS SLB.
Step 8	**exit** **Example:** `Router(config-slb-fw-real)# exit`	Exits real server configuration mode.
Step 9	**access** [**source** *source-ip netmask*] [**destination** *destination-ip netmask*]| **inbound** *inbound-interface* | **outbound** *outbound-interface*] **Example:** `Router(config-slb-fw)# access destination 10.1.6.0 255.255.255.0`	(Optional) Routes specific flows to a firewall farm.
Step 10	**predictor hash address** [**port**] **Example:** `Router(config-slb-fw)# predictor hash address`	(Optional) Specifies whether the source and destination TCP or User Datagram Protocol (UDP) port numbers, in addition to the source and destination IP addresses, are to be used when selecting a firewall.
Step 11	**purge connection**	(Optional) Enables IOS SLB firewall load balancing to send purge requests for connections.

Command or Action	Purpose
Example: `Router(config-slb-fw)# purge connection`	
Step 12 **purge sticky** **Example:** `Router(config-slb-fw)# purge sticky`	(Optional) Enables IOS SLB firewall load balancing to send purge requests when the sticky idle timer expires.

Step 13 **replicate casa** *listen-ip remote-ip port* [*interval*] [**password** [[*encrypt*] *secret-string* [*timeout*]]]
 	
Example:
 	
`Router(config-slb-fw)# replicate casa`
`10.10.10.11 10.10.11.12 4231`
	(Optional) Configures a stateful backup of IOS SLB firewall load balancing decision tables to a backup switch.

Step 14 **protocol tcp** **Example:** `Router(config-slb-fw)# protocol tcp`	(Optional) Enters firewall farm TCP protocol configuration mode.
Step 15 **delay** *duration* **Example:** `Router(config-slb-fw-tcp)# delay 30`	(Optional) For firewall farm TCP protocol configuration mode, specifies the time IOS SLB firewall load balancing maintains TCP connection context after a connection has ended.
Step 16 **idle** *duration* **Example:** `Router(config-slb-fw-tcp)# idle 120`	(Optional) For firewall farm TCP protocol configuration mode, specifies the minimum time IOS SLB firewall load balancing maintains connection context in the absence of packet activity.
Step 17 **maxconns** *maximum-number* **Example:** `Router(config-slb-fw-tcp)# maxconns 1000`	(Optional) For firewall farm TCP protocol configuration mode, specifies the maximum number of active TCP connections allowed on the firewall farm at one time.
Step 18 **sticky** *seconds* [**netmask** *netmask*] [**source**|**destination**] **Example:** `Router(config-slb-fw-tcp)# sticky 60`	(Optional) For firewall farm TCP protocol configuration mode, specifies that connections from the same IP address use the same firewall if either of the following conditions is met: • As long as any connection between the same pair of IP addresses exists (source and destination sticky). • For a period, defined by *duration* , after the last connection is destroyed.
Step 19 **exit**	Exits firewall farm TCP protocol configuration mode.

Command or Action	Purpose
Example: `Router(config-slb-fw-tcp)# exit`	
Step 20 **protocol datagram** **Example:** `Router(config-slb-fw)# protocol datagram`	(Optional) Enters firewall farm datagram protocol configuration mode.
Step 21 **idle** *duration* **Example:** `Router(config-slb-fw-udp)# idle 120`	(Optional) For firewall farm datagram protocol configuration mode, specifies the minimum time IOS SLB firewall load balancing maintains connection context in the absence of packet activity.
Step 22 **maxconns** *maximum-number* **Example:** `Router(config-slb-fw-udp)# maxconns 1000`	(Optional) For firewall farm datagram protocol configuration mode, specifies the maximum number of active datagram connections allowed on the firewall farm at one time.
Step 23 **sticky** *seconds* [**netmask** *netmask*] [**source**|**destination**]	(Optional) For firewall farm datagram protocol configuration mode, specifies that connections from the same IP address use the same firewall if either of the following conditions is met:
Example: `Router(config-slb-fw-udp)# sticky 60`	• As long as any connection between the same pair of IP addresses exists (source and destination sticky). • For a period, defined by *duration* , after the last connection is destroyed.
Step 24 **exit** **Example:** `Router(config-slb-fw-udp)# exit`	Exits firewall farm datagram protocol configuration mode.
Step 25 **inservice** **Example:** `Router(config-slb-fw)# inservice`	Enables the firewall farm for use by IOS SLB.

How to Verify a Firewall Farm

Perform the following optional task to verify a firewall farm.

SUMMARY STEPS

1. The following **show ip slb reals** command displays the status of firewall farm FIRE1, the associated real servers, and their status:
2. The following **show ip slb firewallfarm**command displays the configuration and status of firewall farm FIRE1:

DETAILED STEPS

Step 1 The following **show ip slb reals** command displays the status of firewall farm FIRE1, the associated real servers, and their status:

Example:

```
Router# show ip slb real
real                farm name       weight   state         conns
-------------------------------------------------------------------
10.1.1.2            FIRE1           8        OPERATIONAL   0
10.1.2.2            FIRE1           8        OPERATIONAL   0
```

Step 2 The following **show ip slb firewallfarm**command displays the configuration and status of firewall farm FIRE1:

Example:

```
Router# show ip slb firewallfarm
firewall farm    hash       state       reals
---------------------------------------------
FIRE1            IPADDR     INSERVICE   2
```

How to Verify Firewall Connectivity

Perform the following optional task to verify firewall connectivity.

To verify that IOS SLB firewall load balancing is configured and operating correctly, perform the following steps:

SUMMARY STEPS

1. Ping the external real servers (the ones outside the firewall) from the IOS SLB firewall load-balancing device.
2. Ping the internal real servers (the ones inside the firewall) from the clients.
3. Use the **show ip slb stats** command to display information about the IOS SLB firewall load-balancing network status:
4. Use the **show ip slb real detail**command to display detailed information about the IOS SLB firewall load-balancing real server status:
5. Use the **show ip slb conns**command to display information about active IOS SLB firewall load-balancing connections:

DETAILED STEPS

Step 1 Ping the external real servers (the ones outside the firewall) from the IOS SLB firewall load-balancing device.

Step 2 Ping the internal real servers (the ones inside the firewall) from the clients.

Step 3 Use the **show ip slb stats** command to display information about the IOS SLB firewall load-balancing network status:

Example:

```
Router# show ip slb stats
```

```
Pkts via normal switching:  0
Pkts via special switching: 0
Pkts dropped:               0
Connections Created:        1911871
Connections Established:    1967754
Connections Destroyed:      1313251
Connections Reassigned:     0
Zombie Count:               0
Connections Reused:         59752
Connection Flowcache Purges:1776582
Failed Connection Allocs:   17945
Failed Real Assignments:    0
```

- Normal switching exists when IOS SLB packets are managed on normal IOS switching paths (CEF, fast switching, and process level switching).
- Special switching exists when IOS SLB packets are managed on hardware-assisted switching paths.

Step 4 Use the **show ip slb real detail**command to display detailed information about the IOS SLB firewall load-balancing real server status:

Example:

```
Router# show ip slb reals detail
172.16.88.5, SF1, state = OPERATIONAL, type = server
  ipv6 = 2342:2342:2343:FF04:2388:BB03:3223:8912
  conns = 0, dummy_conns = 0, maxconns = 4294967295
  weight = 8, weight(admin) = 8, metric = 0, remainder = 0
  reassign = 3, retry = 60
  failconn threshold = 8, failconn count = 0
  failclient threshold = 2, failclient count = 0
  total conns established = 0, total conn failures = 0
  server failures = 0
```

Step 5 Use the **show ip slb conns**command to display information about active IOS SLB firewall load-balancing connections:

Example:

```
Router# show ip slb conns
vserver         prot client                real                 state   nat
-----------------------------------------------------------------------------
FirewallTCP     TCP  80.80.50.187:40000    10.1.1.4             ESTAB   none
FirewallTCP     TCP  80.80.50.187:40000    10.1.1.4             ESTAB   none
FirewallTCP     TCP  80.80.50.187:40000    10.1.1.4             ESTAB   none
FirewallTCP     TCP  80.80.50.187:40000    10.1.1.4             ESTAB   none
FirewallTCP     TCP  80.80.50.187:40000    10.1.1.4             ESTAB   none
```

For additional commands used to verify IOS SLB networks and connections, see "How to Monitor and Maintain the Cisco IOS SLB Feature".

How to Configure a Probe

Perform the following required task to configure a probe.

The Exchange Director uses probes to detect and recover from failures. You must configure a probe on each real server in the firewall farm.

- We recommend ping probes for each real server in a firewall farm. For more details, see "How to Configure a Ping Probe".
- If a firewall does not allow ping probes to be forwarded, use HTTP probes instead. For more details, see "How to Configure an HTTP Probe".

- You can configure more than one probe, in any combination of supported types (DNS, HTTP, TCP, or ping), for each firewall in a firewall farm.

How to Configure a Wildcard Search

How to Configure Protocol-Level Purging of MLS entries

To disable purge throttling on TCP and UDP flow packets, use the **no** form of this command.

How to Configure Connection Purge Request Behavior

To completely stop the sending of purge requests, use the **no** form of this command.

How to Configure Sticky Connection Purge Request Behavior

To completely stop the sending of purge requests for sticky connections, use the **no** form of this command.

VPN Server Load Balancing Configuration Task List

SUMMARY STEPS

1. Configure a server farm and a real server.
2. Configure a virtual server.
3. Configure a probe.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Configure a server farm and a real server.	See "How to Configure a Server Farm and a Real Server".
		When you configure the server farm and real server for VPN server load balancing, specify the IP addresses of the real servers acting as VPN terminators using the **real** command.
Step 2	Configure a virtual server.	See "How to Configure a Virtual Server".
		When you configure the virtual server for VPN server load balancing of IPSec flows, keep the following considerations in mind:
		• Configure a UDP virtual server using the **virtual** command with the protocol set to **udp** and the port set to **isakmp**. The **isakmp** keyword enables the cryptographic key exchange to occur through IKE (port 500).
		• Configure an ESP virtual server using the **virtual** command with the protocol set to **esp**.
		• Specify a sticky connection from the UDP virtual server to the ESP virtual server, and vice versa, using the **sticky** command with a *duration* of at least 15 seconds.
		When you configure the virtual server for VPN server load balancing of Point-to-Point Tunneling Protocol (PPTP) flows, keep the following considerations in mind:

Command or Action	Purpose
	• Configure a TCP virtual server, using the **virtual** command with the **tcp** keyword and port number **1723** specified. • Configure a GRE virtual server, using the **virtual** command with the **gre** keyword specified. • Specify a sticky connection from the TCP virtual server to the GRE virtual server, and vice versa, using the **sticky** command with a *duration* of at least 15 seconds.
Step 3 Configure a probe.	See "How to Configure a Probe". To verify the health of the server, configure a ping probe.

ASN Load Balancing Configuration Task List

Perform the following task to configure load balancing across a set of Access Service Network (ASN) gateways.

SUMMARY STEPS

1. Configure the base station.
2. Configure a probe.
3. Associate a server farm and a real server with the probe.
4. Associate a virtual server with the server farm.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Configure the base station.	To enable IOS SLB to manage requests from the Mobile Subscriber Station (MSS), configure the base station with the virtual IP address of the IOS SLB device.
Step 2	Configure a probe.	See "How to Configure a Probe". To verify the health of the server, configure a ping probe.
Step 3	Associate a server farm and a real server with the probe.	See "How to Configure a Server Farm and a Real Server". When you configure the server farm and real server for ASN load balancing, keep the following considerations in mind: • Specify the IP addresses of the ASN gateways, using the **real** command. • (Optional) Enable IOS SLB to automatically remove objects associated with failed real servers from the ASN sticky database, using the **asn purge** option on the **real** command.
Step 4	Associate a virtual server with the server farm.	See "How to Configure a Virtual Server". When you configure the virtual server for ASN load balancing, keep the following considerations in mind: • Configure a virtual server, using the **virtual** command with the service set to **asn**. • Configure an idle connection timer for ASN load balancing, using the **idle** command with the **asn request** keywords specified. • (Optional) Enable IOS SLB to load-balance ASN sessions for a given MSID, using the **asn msid** option on the **sticky** command.

Command or Action	Purpose
	• (Optional) Configure a timer for the ASN MSID sticky database, using the **idle** command with the **asn msid** keywords specified.
	• (Optional) Configure a Cisco BWG port, using the **gw port** command.

Home Agent Director Configuration Task List

Perform the following task to configure the Home Agent Director.

SUMMARY STEPS

1. Configure a server farm and a real server.
2. Configure a virtual server.
3. Configure the virtual IP address as a loopback on each of the home agents in the servers.
4. Configure DFP.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Configure a server farm and a real server.	See "How to Configure a Server Farm and a Real Server".
		When you configure the server farm and real server for the Home Agent Director, keep the following considerations in mind:
		• Accept the default setting (the weighted round robin algorithm) for the **predictor** command.
		• Specify the IP addresses of the real servers acting as home agents, using the **real** command.
Step 2	Configure a virtual server.	See "How to Configure a Virtual Server".
		When you configure the virtual server for the Home Agent Director using the **virtual** command, keep the following considerations in mind:
		• Specify the Home Agent Director's IP address as the virtual server.
		• Specify the **udp** keyword option.
		• Specify port number 434 if the home agents are in compliance with the IP Mobility Support, RFC 2002, 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).
		• Specify the **service ipmobile** keyword option.
Step 3	Configure the virtual IP address as a loopback on each of the home agents in the servers.	(Required for dispatched mode) This step is required only if you are using dispatched mode. Refer to the "Configuring a Loopback Interface" section in the *Cisco IOS Interface Configuration Guide* , Release 12.2 for more information.
Step 4	Configure DFP.	(Optional) See "How to Configure DFP".
		When you configure DFP for the Home Agent Director, keep the following considerations in mind:

Command or Action	Purpose
	• To control the maximum DFP weight sent by the home agent to IOS SLB, use the **ip mobile home-agent dfp-max-weight** command. • To set the source address and home agent address field in the Registration Reply (RRP) as the real home agent's address, use the **ip mobile home-agent dynamic-address**command. • To set the maximum number of bindings, use the **ip mobile home-agent max-binding** command. For information about these Mobile IP commands, refer to the *Cisco Mobile Wireless Home Agent Release 2.0* feature module.

How to Configure NAT

Perform the following task to configure the IOS SLB Network Address Translation (NAT) client address pool for client NAT.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb natpool pool** *start-ip end-ip* [**netmask** *netmask* | **prefix-length** *leading-1-bits*] [**entries** *init-address* [*max-address*]]
4. **nat** {**client** *pool* | **server**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip slb natpool pool** *start-ip end-ip* [**netmask** *netmask*	**prefix-length** *leading-1-bits*] [**entries** *init-address* [*max-address*]] **Example:** `Router(config)# ip slb natpool web-clients 10.1.10.1 10.1.10.5 netmask 255.255.0.0`	Configures the client address pool. GPRS load balancing does not support this command. You do not need to configure the client address pool for server NAT.

	Command or Action	Purpose
Step 4	**nat** {**client** *pool* | **server**} **Example:** `Router(config-slb-sfarm)# nat server`	Configures SLB NAT and specifies a NAT mode. All IPv4 or IPv6 server farms that are associated with the same virtual server must have the same NAT configuration.

You must also specify either NAT client translation mode or NAT server address translation mode on the server farm, using the **nat**command. See "How to Configure a Server Farm and a Real Server" for more details. When you configure the virtual server for NAT, remember that you cannot configure client NAT for an ESP or GRE virtual server.

How to Configure Static NAT

Perform the following task to configure static NAT.

Static NAT enables you to allow some users to use NAT and allow other users on the same Ethernet interface to continue with their own IP addresses. This option enables you to provide a default NAT behavior for real servers, differentiating between responses from a real server, and connection requests initiated by the real server.

Note To avoid unexpected results, make sure your static NAT configuration mirrors your virtual server configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb static** {**drop** \| **nat** {**virtual** \| *virtual-ip*[**per-packet** \| **sticky**]}}
4. **real** *ip-address* [*port*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip slb static** {**drop** | **nat** {**virtual** | *virtual-ip*[**per-packet** | **sticky**]}} **Example:** `Router(config)# ip slb static nat 10.1.10.1 per-packet`	Configures the real server's NAT behavior and enters static NAT configuration mode. **Note** If you specify the *virtual-ip* argument and you do not specify the **per-packet** option, IOS SLB uses server port translation to distinguish between connection requests initiated by different real servers.
Step 4	**real** *ip-address* [*port*] **Example:** `Router(config-slb-static)# real 10.1.1.3`	Configures one or more real servers to use static NAT.

Stateless Backup Configuration Task List

Perform the following task to configure stateless backup over VLANs between IOS SLB devices.

Note For active standby, in which multiple IOS SLB devices share a virtual IP address, you must use exclusive client ranges and you must use policy routing to forward flows to the correct IOS SLB device.

SUMMARY STEPS

1. Configure required and optional IOS SLB functions.
2. Configure firewall load balancing.
3. Configure the IP routing protocol.
4. Configure the VLAN between the IOS SLB devices.
5. Verify the stateless backup configuration.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Configure required and optional IOS SLB functions.	(Required for server load balancing) See "How to Configure Required and Optional IOS SLB Functions".
Step 2	Configure firewall load balancing.	(Required for firewall load balancing) See "How to Configure Firewall Load Balancing".
Step 3	Configure the IP routing protocol.	Refer to the "IP Routing Protocols" chapter of the *Cisco IOS IP Configuration Guide*, Release 12.2 for details.
Step 4	Configure the VLAN between the IOS SLB devices.	Refer to the "Virtual LANs" chapter of the *Cisco IOS Switching Services Configuration Guide*, Release 12.2 for details.
Step 5	Verify the stateless backup configuration.	(Optional) See "How to Verify the Stateless Backup Configuration".

How to Verify the Stateless Backup Configuration

Perform the following task to verify the stateless backup configuration.

For server load balancing, to verify that stateless backup has been configured and is operating correctly, use the following **show ip slb vservers** commands to display information about the IOS SLB virtual server status:

```
Router# show ip slb vservers
slb vserver       prot  virtual               state         conns
-----------------------------------------------------------------
VS1               TCP   10.10.10.12:23        OPERATIONAL   2
VS2               TCP   10.10.10.18:23        OPERATIONAL   2
Router# show ip slb vservers detail
VS1, state = OPERATIONAL, v_index = 10
  virtual = 10.10.10.12:23, TCP, service = NONE, advertise = TRUE
  server farm = SERVERGROUP1, delay = 10, idle = 3600
  sticky timer = 0, sticky subnet = 255.255.255.255
  sticky group id = 0
  synguard counter = 0, synguard period = 0
  conns = 0, total conns = 0, syns = 0, syn drops = 0
  standby group = None
VS2, state = INSERVICE, v_index = 11
  virtual = 10.10.10.18:23, TCP, service = NONE, advertise = TRUE
  server farm = SERVERGROUP2, delay = 10, idle = 3600
  sticky timer = 0, sticky subnet = 255.255.255.255
  sticky group id = 0
  synguard counter = 0, synguard period = 0
  conns = 0, total conns = 0, syns = 0, syn drops = 0
  standby group = None
```

For firewall load balancing, to verify that stateless backup has been configured and is operating correctly, use the following **show ip slb firewallfarm**commands to display information about the IOS SLB firewall farm status:

```
Router# show ip slb firewallfarm

firewall farm    hash        state         reals
------------------------------------------------
FIRE1            IPADDR      INSERVICE     2
Router# show ip slb firewallfarm details
FIRE1, hash = IPADDRPORT, state = INSERVICE, reals = 2
  FirewallTCP:
   sticky timer = 0, sticky subnet = 255.255.255.255
   idle = 3600, delay = 10, syns = 1965732, syn drop = 0
   maxconns = 4294967295, conns = 597445, total conns = 1909512
  FirewallUDP:
   sticky timer = 0, sticky subnet = 255.255.255.255
   idle = 3600
   maxconns = 1, conns = 0, total conns = 1
  Real firewalls:
    10.1.1.3, weight = 10, OPERATIONAL, conns = 298823
    10.1.1.4, weight = 10, OPERATIONAL, conns = 298622
  Total connections = 597445
```

Stateful Backup of Redundant Route Processors Configuration Task List

SUMMARY STEPS

1. Configure the replication message rate for slave replication.
2. Configure required and optional IOS SLB functions.
3. Configure firewall load balancing.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Configure the replication message rate for slave replication.	Specify the **ip slb replicate slave rate** command in global configuration mode.
Step 2	Configure required and optional IOS SLB functions.	(Required for server load balancing) See "How to Configure Required and Optional IOS SLB Functions". When you configure the virtual server for stateful backup of redundant route processors, keep the following considerations in mind: • Specify the **replicate slave** command. • (Optional) To set the replication delivery interval for the virtual server, configure a **replicate interval** command.
Step 3	Configure firewall load balancing.	(Required for firewall load balancing) See "How to Configure Firewall Load Balancing". When you configure the firewall farm for stateful backup of redundant route processors, keep the following considerations in mind: • Specify the **replicate slave** command. • (Optional) To set the replication delivery interval for the firewall farm, configure a **replicate interval** command.

How to Configure Database Entries

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb entries** [**conn** [*init-conn* [*max-conn*]] | **frag** [*init-frag* [*max-frag*] | **lifetime** *timeout*] | **gtp** {**gsn** [*init-gsn*[*max-gsn*] | **nsapi** [*init-nsapi* [*max-nsapi*]} | **sticky** [*init-sticky* [*max-sticky*]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. If prompted, enter your password.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip slb entries** [**conn** [*init-conn* [*max-conn*]] | **frag** [*init-frag* [*max-frag*]] | **lifetime** *timeout*] | **gtp** {**gsn** [*init-gsn*[*max-gsn*] | **nsapi** [*init-nsapi* [*max-nsapi*]} | **sticky** [*init-sticky* [*max-sticky*]]] **Example:** `Router(config)# ip slb entries conn 128000 512000`	Specifies an initial allocation and a maximum value for IOS SLB database entries. **Note** Enter this command *before* entering the rest of your IOS SLB configuration. If your IOS SLB configuration already exists, you must reload ISO SLB after entering this command.

How to Configure Buffers for the Fragment Database

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb maxbuffers frag** *buffers*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip slb maxbuffers frag** *buffers* **Example:** `Router(config)# ip slb maxbuffers frag 300`	Configures the maximum number of buffers for the IOS SLB fragment database.

How to Clear Databases and Counters

SUMMARY STEPS

1. **clear ip slb connections** [**firewallfarm** *firewall-farm*| **serverfarm** *server-farm*| **vserver** *virtual-server*]
2. **clear ip slb counters** [**kal-ap**]
3. **clear ip slb sessions** [**firewallfarm** *firewall-farm*| **serverfarm** *server-farm* | **vserver** *virtual-server*]
4. **clear ip slb sticky asn msid** *msid*
5. **clear ip slb sticky gtp imsi** [**id** *imsi*]
6. **clear ip slb sticky radius** {**calling-station-id** [**id** *string*] | framed-ip [*framed-ip* [*netmask*]]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**clear ip slb connections** [**firewallfarm** *firewall-farm*| **serverfarm** *server-farm*| **vserver** *virtual-server*] **Example:** `Router# clear ip slb connections vserver VSERVER1`	Clears the IOS SLB connection database for one or more firewall farms, server farms, or virtual servers.
Step 2	**clear ip slb counters** [**kal-ap**] **Example:** `Router# clear ip slb counters`	Clears the IOS SLB counters. Use the **kal-ap** keyword to clear only IP IOS SLB KeepAlive Application Protocol (KAL-AP) counters.
Step 3	**clear ip slb sessions** [**firewallfarm** *firewall-farm*| **serverfarm** *server-farm* | **vserver** *virtual-server*] **Example:** `Router# clear ip slb sessions serverfarm FARM1`	Clears the IOS SLB RADIUS session database for one or more firewall farms, server farms, or virtual servers.
Step 4	**clear ip slb sticky asn msid** *msid* **Example:** `Router# clear ip slb sticky asn msid 001646013fc0`	Clears entries from an IOS SLB Access Service Network (ASN) Mobile Station ID (MSID) sticky database.
Step 5	**clear ip slb sticky gtp imsi** [**id** *imsi*] **Example:** `Router# clear ip slb sticky gtp imsi`	Clears entries from an IOS SLB general packet radio service (GPRS) Tunneling Protocol (GTP) International Mobile Subscriber ID (IMSI) sticky database.
Step 6	**clear ip slb sticky radius** {**calling-station-id** [**id** *string*] | framed-ip [*framed-ip* [*netmask*]]} **Example:** `Router# clear ip slb sticky radius framed-ip`	Clears entries from an IOS SLB RADIUS sticky database.

How to Configure a Wildcard Search

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Router(config)# **mls ip slb search**{**wildcard** [**pfc** | **rp**] | **icmp**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Router(config)# **mls ip slb search**{**wildcard** [**pfc**	**rp**]	**icmp**} **Example:** `Router(config)# mls ip slb search wildcard rp`	Specifies the behavior of IOS SLB wildcard searches. This command is supported for Cisco Catalyst 6500 series switch only.

How to Configure Protocol-Level Purging of MLS Entries

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Router(config)# **mls ip slb purge global**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal**	Enters global configuration mode.

Command or Action	Purpose
Example: `Router# configure terminal`	
Step 3 Router(config)# **mls ip slb purge global** **Example:** `Router(config)# mls ip slb purge global`	Specifies protocol-level purging of MLS entries from active TCP and UDP flow packets. This command is supported for Cisco Catalyst 6500 series switches only.

How to Purge and Reassign Connections

You can enable IOS SLB to automatically remove connections to failed real servers and firewalls from the connection database even if the idle timers have not expired. This function is useful for applications that do not rotate the source port (such as IKE), and for protocols that do not have ports to differentiate flows (such as ESP).

You can also enable IOS SLB to automatically reassign to a new real server or firewall RADIUS sticky objects that are destined for a failed real server or firewall.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb serverfarm** *server-farm*
4. **failaction** [**purge** | **asn purge**| **gtp purge**| **radius reassign**]
5. **exit**
6. **ip slb firewallfarm** *firewall-farm*
7. **failaction purge**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip slb serverfarm** *server-farm*	Enters server farm configuration mode.

Command or Action	Purpose
Example: `Router(config)# ip slb serverfarm PUBLIC`	
Step 4 **failaction** [**purge** | **asn purge**| **gtp purge**| **radius reassign**] **Example:** `Router(config-slb-sfarm)# failaction purge`	Configures IOS SLB behavior in the event that a real server fails.
Step 5 **exit** **Example:** `Router(config-slb-sfarm)# exit`	Exits server farm configuration mode.
Step 6 **ip slb firewallfarm** *firewall-farm* **Example:** `Router(config)# ip slb firewallfarm fire1`	Enters firewall farm configuration mode.
Step 7 **failaction purge** **Example:** `Router(config-slb-fw)# failaction purge`	Configures IOS SLB behavior in the event that a firewall fails.

How to Disable Automatic Server Failure Detection

If you have configured all-port virtual servers (that is, virtual servers that accept flows destined for all ports except GTP ports), flows can be passed to servers for which no application port exists. When the servers reject these flows, IOS SLB might fail the servers and remove them from load balancing. This situation can also occur in slow-to-respond AAA servers in RADIUS load-balancing environments. To prevent this situation, you can disable automatic server failure detection.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip slb serverfarm** *server-farm*
4. **real** *ipv4-address* [**ipv6** *ipv6-address*] [*port*]
5. **no faildetect inband**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**ip slb serverfarm** *server-farm* **Example:** Router(config)# ip slb serverfarm PUBLIC	Enters server farm configuration mode.
Step 4	**real** *ipv4-address* [**ipv6** *ipv6-address*] [*port*] **Example:** Router(config-slb-sfarm)# real 10.1.1.1	Identifies a real server as a member of a server farm and enters real server configuration mode. **Note** For dual-stack support for GTP load balancing, specify the real server's IPv4 and IPv6 address.
Step 5	**no faildetect inband** **Example:** Router(config-slb-real)# no faildetect inband	Disables automatic server failure detection. **Note** If you disable automatic server failure detection using the **no faildetect inband** command, We recommend that you configure one or more probes. If you specify the **no faildetect inband** command, the **faildetect numconns** command is ignored, if specified.

How to Monitor and Maintain the Cisco IOS SLB Feature

Perform the following task to obtain and display runtime information about IOS SLB.

SUMMARY STEPS

1. **show ip slb conns** [**vserver** *virtual-server* | **client** *ip-address* | **firewall** *firewall-farm*] [**detail**]
2. **show ip slb dfp** [**agent** *agent-ip port* | **manager** *manager-ip* | **detail** | **weights**]
3. **show ip slb firewallfarm** [**detail**]
4. **show ip slb fragments**
5. **show ip slb gtp** {**gsn** [*gsn-ip-address*] | **nsapi** [*nsapi-key*] [**detail**]
6. **show ip slb map** [*map-id*]
7. **show ip slb natpool** [**name** *pool*] [**detail**]
8. **show ip slb probe** [**name** *probe*] [**detail**]
9. **show ip slb reals** [**sfarm** *server-farm*] [**detail**]
10. **show ip slb replicate**
11. **show ip slb serverfarms** [**name** *server-farm*] [**detail**]
12. **show ip slb sessions** [**asn**| **gtp**[**ipv6**] | **gtp-inspect**| **ipmobile**| **radius**] [**vserver** *virtual-server*] [**client** *ipv4-address netmask*] [**detail**]
13. **show ip slb static**
14. **show ip slb stats**
15. **show ip slb sticky** [**client** *ip-address netmask*| **radius calling-station-id**[**id** *string*] | **radius framed-ip**[**client** *ip-address netmask*] | **radius username**[**name** *string*]]
16. **show ip slb vservers** [**name** *virtual-server*] [**redirect**] [**detail**]
17. **show ip slb wildcard**

DETAILED STEPS

Step 1　**show ip slb conns** [**vserver** *virtual-server* | **client** *ip-address* | **firewall** *firewall-farm*] [**detail**]

Displays all connections managed by IOS SLB, or, optionally, only those connections associated with a particular virtual server or client. The following is sample output from this command:

Example:

```
Router# show ip slb conns
vserver          prot   client                real                      state
--------------------------------------------------------------------------
TEST             TCP    10.150.72.183:328     10.80.90.25:80            INIT
TEST             TCP    10.250.167.226:423    10.80.90.26:80            INIT
TEST             TCP    10.234.60.239:317     10.80.90.26:80            ESTAB
TEST             TCP    10.110.233.96:747     10.80.90.26:80            ESTAB
TEST             TCP    10.162.0.201:770      10.80.90.30:80            CLOSING
TEST             TCP    10.22.225.219:995     10.80.90.26:80            CLOSING
TEST             TCP    10.2.170.148:169      10.80.90.30:80
```

Step 2　**show ip slb dfp** [**agent** *agent-ip port* | **manager** *manager-ip* | **detail** | **weights**]

Displays information about Dynamic Feedback Protocol (DFP) and DFP agents, and about the weights assigned to real servers. The following is sample output from this command:

Example:

```
Router# show ip slb dfp
DFP Manager:
Current passwd:NONE Pending passwd:NONE
Passwd timeout:0 sec
```

```
Agent IP          Port    Timeout  Retry Count  Interval
----------------------------------------------------------
172.16.2.34       61936   0        0            180 (Default)
```

Step 3   **show ip slb firewallfarm** [**detail**]

Displays information about firewall farms. The following is sample output from this command:

Example:

```
Router# show ip slb firewallfarm
firewall farm    hash        state         reals
-------------------------------------------------
FIRE1            IPADDR      OPERATIONAL   2
```

Step 4   **show ip slb fragments**

Displays information from the IOS SLB fragment database. The following is sample output from this command:

Example:

```
Router# show ip slb fragments
ip src           id    forward         src nat         dst nat
----------------------------------------------------------------------
10.11.2.128      12    10.11.2.128     10.11.11.11     10.11.2.128
10.11.2.128      13    10.11.2.128     10.11.11.11     10.11.2.128
10.11.2.128      14    10.11.2.128     10.11.11.11     10.11.2.128
10.11.2.128      15    10.11.2.128     10.11.11.11     10.11.2.128
10.11.2.128      16    10.11.2.128     10.11.11.11     10.11.2.128
```

Step 5   **show ip slb gtp** {**gsn** [*gsn-ip-address*] | **nsapi** [*nsapi-key*] [**detail**]

Displays IOS SLB GPRS Tunneling Protocol (GTP) information. The following is sample output from this command:

Example:

```
Router# show ip slb gtp gsn 10.0.0.0
type ip                recovery-ie  purging
------------------------------------------
SGSN 10.0.0.0 UNKNOWN       N
```

Step 6   **show ip slb map** [*map-id*]

Displays information about IOS SLB protocol maps. The following is sample output from this command:

Example:

```
Router# show ip slb map
ID: 1, Service: GTP
 APN: Cisco.com, yahoo.com
 PLMN ID(s): 11122, 444353
 SGSN access list: 100
ID: 2, Service: GTP
 PLMN ID(s): 67523, 345222
 PDP Type: IPv4, PPP
ID: 3, Service: GTP
 PDP Type: IPv6
ID: 4, Service: RADIUS
 Calling-station-id: "?919*"
ID: 5, Service: RADIUS
 Username: ". .778cisco.*"
```

Step 7   **show ip slb natpool** [**name** *pool*] [**detail**]

Displays information about the IOS SLB NAT configuration. The following is sample output from this command:

Example:

```
Router# show ip slb natpool
```

```
nat client B 209.165.200.225 1.1.1.6 1.1.1.8 Netmask 255.255.255.0
nat client A 10.1.1.1 1.1.1.5 Netmask 255.255.255.0
```

Step 8   **show ip slb probe** [**name** *probe*] [**detail**]

Displays information about probes defined to IOS SLB. The following is sample output from this command:

Example:

```
Router# show ip slb probe
Server:Port          State        Outages  Current  Cumulative
-------------------------------------------------------------
10.10.4.1:0          OPERATIONAL      0  never    00:00:00
10.10.5.1:0          FAILED           1  00:00:06 00:00:06
```

Step 9   **show ip slb reals** [**sfarm** *server-farm*] [**detail**]

Displays information about the real servers defined to IOS SLB. The following is sample output from this command:

Example:

```
Router# show ip slb reals
real            farm name      weight  state         conns
------------------------------------------------------------
10.80.2.112     FRAG           8       OUTOFSERVICE  0
10.80.5.232     FRAG           8       OPERATIONAL   0
10.80.15.124    FRAG           8       OUTOFSERVICE  0
10.254.2.2      FRAG           8       OUTOFSERVICE  0
10.80.15.124    LINUX          8       OPERATIONAL   0
10.80.15.125    LINUX          8       OPERATIONAL   0
10.80.15.126    LINUX          8       OPERATIONAL   0
10.80.90.25     SRE            8       OPERATIONAL   220
10.80.90.26     SRE            8       OPERATIONAL   216
10.80.90.27     SRE            8       OPERATIONAL   216
10.80.90.28     SRE            8       TESTING       1
10.80.90.29     SRE            8       OPERATIONAL   221
10.80.90.30     SRE            8       OPERATIONAL   224
10.80.30.3      TEST           100     READY_TO_TEST 0
10.80.30.4      TEST           100     READY_TO_TEST 0
10.80.30.5      TEST           100     READY_TO_TEST 0
10.80.30.6      TEST           100     READY_TO_TEST 0
```

Step 10   **show ip slb replicate**

Displays information about the IOS SLB replication configuration. The following is sample output from this command:

Example:

```
Router# show ip slb replicate

VS1, state = NORMAL, interval = 10
 Slave Replication: Enabled
 Slave Replication statistics:
  unsent conn updates:         0
  conn updates received:       0
  conn updates transmitted:    0
  update messages received:    0
  update messages transmitted: 0
 Casa Replication:
  local = 10.1.1.1 remote = 10.2.2.2 port = 1024
  current password = <none> pending password = <none>
  password timeout = 180 sec (Default)
 Casa Replication statistics:
  unsent conn updates:         0
  conn updates received:       0
  conn updates transmitted:    0
  update packets received:     0
  update packets transmitted: 0
  failovers:                   0
```

Step 11   **show ip slb serverfarms** [**name** *server-farm*] [**detail**]

Displays information about the server farms defined to IOS SLB. The following is sample output from this command:

Example:

```
Router# show ip slb serverfarms
server farm      predictor      reals   bind id
-------------------------------------------------
FRAG             ROUNDROBIN     4       0
LINUX            ROUNDROBIN     3       0
SRE              ROUNDROBIN     6       0
TEST             ROUNDROBIN     4       0
```

Step 12   **show ip slb sessions** [**asn**| **gtp**[**ipv6**] | **gtp-inspect**| **ipmobile**| **radius**] [**vserver** *virtual-server*] [**client** *ipv4-address netmask*] [**detail**]

Displays information about sessions managed by IOS SLB. The following is sample output from this command:

Example:

```
Router# show ip slb sessions radius
Source                 Dest                     Retry
Addr/Port              Addr/Port         Id Count  Real          Vserver
--------------------------------------------------------------------------
10.10.11.1/1645      10.10.11.2/1812    15    1  10.10.10.1  RADIUS_ACCT
```

Step 13   **show ip slb static**

Displays information about the IOS SLB server Network Address Translation (NAT) configuration. The following is sample output from this command:

Example:

```
Router# show ip slb static
real                   action        address        counter
-----------------------------------------------------------------
10.11.3.4              drop          0.0.0.0        0
10.11.3.1              NAT           10.11.11.11    3
10.11.3.2              NAT sticky    10.11.11.12    0
10.11.3.3              NAT per-packet 10.11.11.13   0
```

Step 14   **show ip slb stats**

Displays IOS SLB statistics. The following is sample output from this command:

Example:

```
Router# show ip slb stats
Pkts via normal switching:     779
Pkts via special switching:    0
Pkts via slb routing:          0
Pkts Dropped:                  4
Connections Created:           4
Connections Established:       4
Connections Destroyed:         4
Connections Reassigned:        5
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
RADIUS calling-station-id Sticky Count:  0
GTP IMSI Sticky Count:         0
Failed Correlation Injects:    0
Pkt fragments drops in ssv:    0
ASN MSID sticky count:         1
```

Step 15   **show ip slb sticky** [**client** *ip-address netmask*| **radius calling-station-id**[**id** *string*] | **radius framed-ip**[**client** *ip-address netmask*] | **radius username**[**name** *string*]]

Displays information about the sticky connections defined to IOS SLB. The following is sample output from this command:

Example:

```
Router# show ip slb sticky
client          netmask          group real                conns
----------------------------------------------------------------------
10.10.2.12      255.255.0.0      4097  10.10.3.2           1
```

Step 16   **show ip slb vservers** [**name** *virtual-server*] [**redirect**] [**detail**]
Displays information about the virtual servers defined to IOS SLB. The following is sample output from this command:

Example:

```
Router# show ip slb vservers
slb vserver      prot   virtual            state          conns
--------------------------------------------------------------------
TEST             TCP    10.80.254.3:80     OPERATIONAL    1013
TEST21           TCP    10.80.254.3:21     OUTOFSERVICE   0
TEST23           TCP    10.80.254.3:23     OUTOFSERVICE   0
```

Step 17   **show ip slb wildcard**
Displays information about the wildcard representation for virtual servers defined to IOS SLB. The following is sample output from this command:

Example:

```
Router# show ip slb wildcard
Interface Source Address      Port  Destination Address   Port  Prot
ANY       0.0.0.0/0           0     3.3.3.3/32            2123  UDP
ANY       0.0.0.0/0           0     3.3.3.3/32            0     UDP
ANY       0.0.0.0/0           0     0.0.0.0/0            0     ICMP
Interface: ANY
Source Address [Port]: : :/0[0]
Destination Address [Port]: 2342:2342:2343:FF04:2341:AA03:2323:8912/128[0]
Protocol: ICMPV6
Interface: ANY
Source Address [Port]: : :/0[0]
Destination Address [Port]: 2342:2342:2343:FF04:2341:AA03:2323:8912/128[2123]
Protocol: UDP
```

Configuration Examples for IOS SLB

This section provides real-world examples of IOS SLB configurations. For a complete description of the IOS SLB commands in this section, refer to the *Cisco IOS IP Application Services Command Reference*. To locate documentation of other commands that appear in this section, search online using Cisco.com.

This section includes the following examples:

Note The IP and network addresses in these examples are generic; you must replace them with the actual addresses for your network.

Example How to Configure a Basic IOS SLB Network

The figure below shows a sample IOS SLB network with the following components:

- Two server farms--one configured to allow access by the public and named PUBLIC, one configured to allow limited access and named RESTRICTED.
- Five real servers configured as follows:
  - Three real servers in the PUBLIC server farm with IP addresses 10.1.1.1, 10.1.1.2, and 10.1.1.3
  - Two real servers in the restricted server farm with IP addresses 10.1.1.20 and 10.1.1.21
- Two virtual servers--one configured to allow access by the public and named PUBLIC_HTTP and one configured to allow limited access and named RESTRICTED_HTTP.
  - Virtual server PUBLIC_HTTP is configured with IP address 10.0.0.1 load balancing TCP connections on the WWW port (80).
  - Virtual server RESTRICTED_HTTP is configured with IP address 10.0.0.2 load balancing TCP connections on the WWW port (80) and allows access only from clients from network 10.4.4.0 255.255.255.0.

Figure 2: Example IOS SLB Network
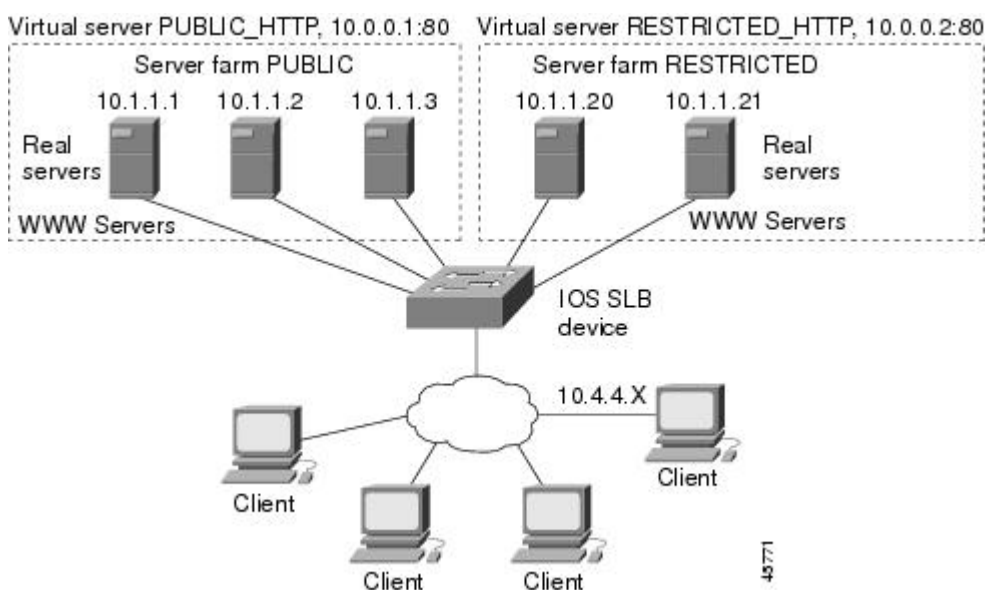
The following sections include examples of the configuration commands used to configure and verify the IOS SLB network shown in the figure above.

Server Farm Configuration

The following example shows the configuration for the server farm PUBLIC, associated with three real servers:

```
ip slb serverfarm PUBLIC
  real 10.1.1.1
    reassign 2
    faildetect numconns 4 numclients 2
    retry 20
```

```
      inservice
      exit
   real 10.1.1.2
     reassign 2
     faildetect numconns 4
     retry 20
     inservice
     exit
   real 10.1.1.3
     reassign 2
     faildetect numconns 4
     retry 20
     inservice
     end
```

The following example shows the configuration for the server farm RESTRICTED, associated with two real servers:

```
ip slb serverfarm RESTRICTED
  real 10.1.1.20
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
    exit
  real 10.1.1.21
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
    end
```

Virtual Server Configuration

The following example shows the configuration for the virtual servers PUBLIC_HTTP and RESTRICTED_HTTP:

```
ip slb vserver PUBLIC_HTTP
  virtual 10.0.0.1 tcp www
  serverfarm PUBLIC
  idle 120
  delay 5
  inservice
  exit
ip slb vserver RESTRICTED_HTTP
  virtual 10.0.0.2 tcp www
  serverfarm RESTRICTED
  idle 120
  delay 5
  inservice
  end
```

Restricted Client Configuration

The following example shows the configuration for the virtual server RESTRICTED_HTTP:

```
ip slb vserver RESTRICTED_HTTP
  no inservice
  client 10.4.4.0 255.255.255.0
  inservice
  end
```

Example How to Configure a Complete IOS SLB Network

The following example provides a complete configuration using many of the commands described in this feature document:

```
ip slb probe PROBE2 http
 request method POST url /probe.cgi?all
 header HeaderName HeaderValue
!
ip slb serverfarm PUBLIC
 nat server
 real 10.1.1.1
  reassign 4
  faildetect numconns 16
  retry 120
  inservice
 real 10.1.1.2
  reassign 4
  faildetect numconns 16
  retry 120
  inservice
 probe PROBE2
!
ip slb serverfarm RESTRICTED
 predictor leastconns
 bindid 309
 real 10.1.1.1
  weight 32
  maxconns 1000
  reassign 4
  faildetect numconns 16
  retry 120
  inservice
 real 10.1.1.20
  reassign 4
  faildetect numconns 16
  retry 120
  inservice
 real 10.1.1.21
  reassign 4
  faildetect numconns 16
  retry 120
  inservice
!
ip slb vserver PUBLIC_HTTP
 virtual 10.0.0.1 tcp www
 serverfarm PUBLIC
!
ip slb vserver RESTRICTED_HTTP
 virtual 10.0.0.2 tcp www
 serverfarm RESTRICTED
 no advertise
 sticky 60 group 1
 idle 120
 delay 5
 client 10.4.4.0 255.255.255.0
 synguard 3600000
 inservice
```

Examples How to Configure IOS SLB with Firewall Load Balancing

This section contains the following examples, illustrating several different IOS SLB firewall load-balancing configurations:

Example How to Configure IOS SLB with Basic Firewall Load Balancing

The figure below shows a sample IOS SLB firewall load-balancing network with the following components:

- Two firewalls with IP addresses as shown
- An internal firewall load-balancing device on the secure side of the firewalls
- An external firewall load-balancing device on the Internet side of the firewalls
- One firewall farm named FIRE1, containing both firewalls

Figure 3: IOS SLB with Layer 3 Firewalls in Different Subnets
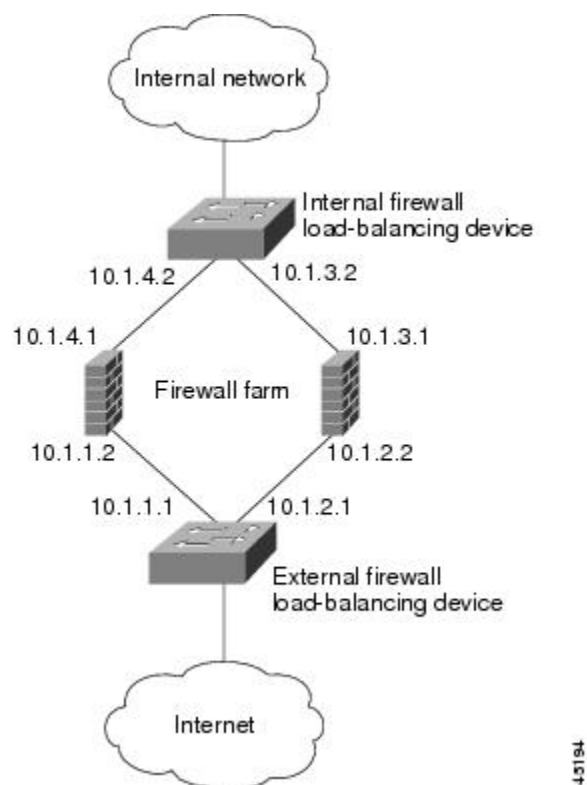
When you configure IOS SLB firewall load balancing, the load-balancing devices use route lookup to recognize flows destined for the firewalls. To enable route lookup, you must configure each device with the IP address of each firewall that will route flows to that device.

In the following firewall farm configuration samples:

- The internal (secure side) firewall load-balancing device is configured with firewall IP addresses 10.1.3.1 and 10.1.4.1.

- The external (Internet side) firewall load-balancing device is configured with firewall IP addresses 10.1.1.2 and 10.1.2.2.

Internal Firewall Load-Balancing Device

The following example shows the configuration for ping probe PROBE1, HTTP probe PROBE2, and firewall farm FIRE1, associated with the two real servers for the load-balancing device on the internal (secure) side of the firewall:

```
!-----Ping probe
ip slb probe PROBE1 ping
!-----IP address of other load-balancing device
  address 10.1.1.1
  faildetect 4
!-----HTTP probe
  ip slb probe PROBE2 http
!-----IP address of other load-balancing device
  address 10.1.2.1
  expect status 401
!-----Firewall farm FIRE1
ip slb firewallfarm FIRE1
!-----First firewall
  real 10.1.4.1
    probe PROBE1
!-----Enable first firewall
    inservice
!-----Second firewall
    real 10.1.3.1
    probe PROBE2
!-----Enable second firewall
    inservice
```

External Firewall Load-Balancing Device

The following example shows the configuration for ping probe PROBE1, HTTP probe PROBE2, and firewall farm FIRE1, associated with the two real servers for the load-balancing device on the external (Internet) side of the firewall:

```
!-----Ping probe
ip slb probe PROBE1 ping
!-----IP address of other load-balancing device
  address 10.1.4.2
  faildetect 4
!-----HTTP probe
ip slb probe PROBE2 http
!-----IP address of other load-balancing device
  address 10.1.3.2
  expect status 401
!-----Firewall farm FIRE1
ip slb firewallfarm FIRE1
!-----First firewall
  real 10.1.1.2
    probe PROBE1
!-----Enable first firewall
    inservice
!-----Second firewall
  real 10.1.2.2
    probe PROBE2
!-----Enable second firewall
    inservice
    exit
  inservice
```

Example How to Configure IOS SLB with Server Load Balancing and Firewall Load Balancing

The figure below shows a sample IOS SLB load-balancing network with server load balancing and firewall load balancing running together, and the following components:

- Two real servers with IP addresses as shown
- One server farm named PUBLIC, containing both real servers
- Two firewalls with IP addresses as shown
- One firewall farm named FIRE1, containing both firewalls
- An internal IOS SLB device on the secure side of the firewalls, performing server load balancing and firewall load balancing
- An external firewall load-balancing device on the Internet side of the firewalls

Figure 4: IOS SLB with Server Load Balancing and Firewall Load Balancing
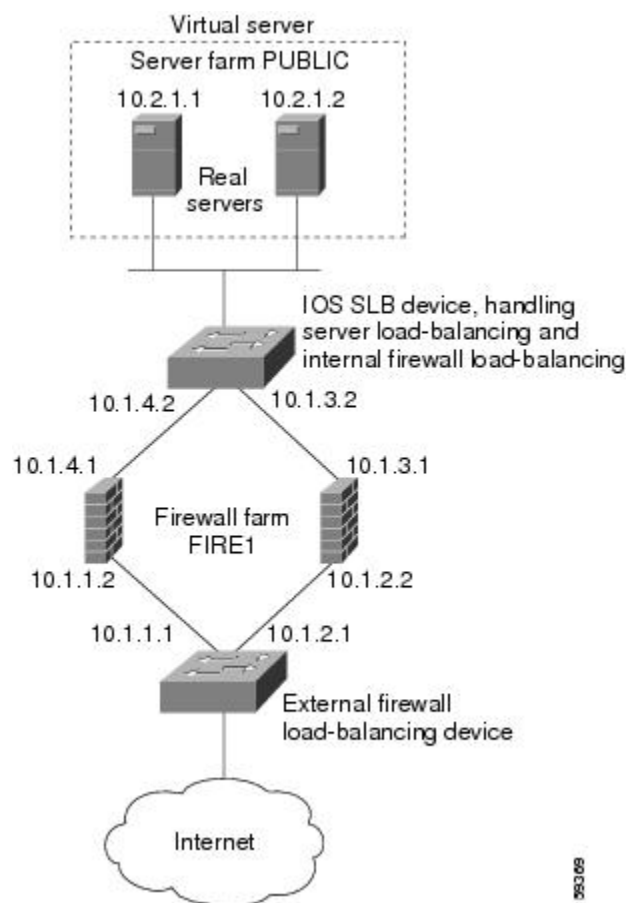
In the following firewall farm configuration samples:

- The internal (secure side) firewall load-balancing device is configured with firewall IP addresses 10.1.3.1 and 10.1.4.1.
- The external (Internet side) firewall load-balancing device is configured with firewall IP addresses 10.1.1.2 and 10.1.2.2.

Internal Server and Firewall Load-Balancing Device

The following example shows the configuration for ping probes ABCPROBE and XYZPROBE, firewall farm FIRE1, and server farm PUBLIC for the load-balancing device on the internal (secure) side of the firewalls:

```
ip slb probe ABCPROBE ping
  address 10.1.1.1
ip slb probe XYZPROBE ping
  address 10.1.2.1
!
ip slb firewallfarm FIRE1
  real 10.1.4.1
    probe ABCPROBE
    inservice
  real 10.1.3.1
    probe XYZPROBE
    inservice
  inservice
!
ip slb serverfarm PUBLIC
  nat server
  real 10.2.1.1
    inservice
    real 10.2.1.2
    inservice
!
ip slb vserver HTTP1
  virtual 128.1.0.1 tcp www
  serverfarm PUBLIC
  idle 120
  delay 5
  inservice
```

External Firewall Load-Balancing Device

The following example shows the configuration for ping probes ABCPROBE and XYZPROBE and firewall farm FIRE1 for the load-balancing device on the external (Internet) side of the firewalls:

```
ip slb probe ABCPROBE ping
  address 10.1.4.2
  ip slb probe XYZPROBE ping
  address 10.1.3.2
  ip slb firewallfarm FIRE1
  real 10.1.1.2
    probe ABCPROBE
    inservice
    probe XYZPROBE
    inservice
```
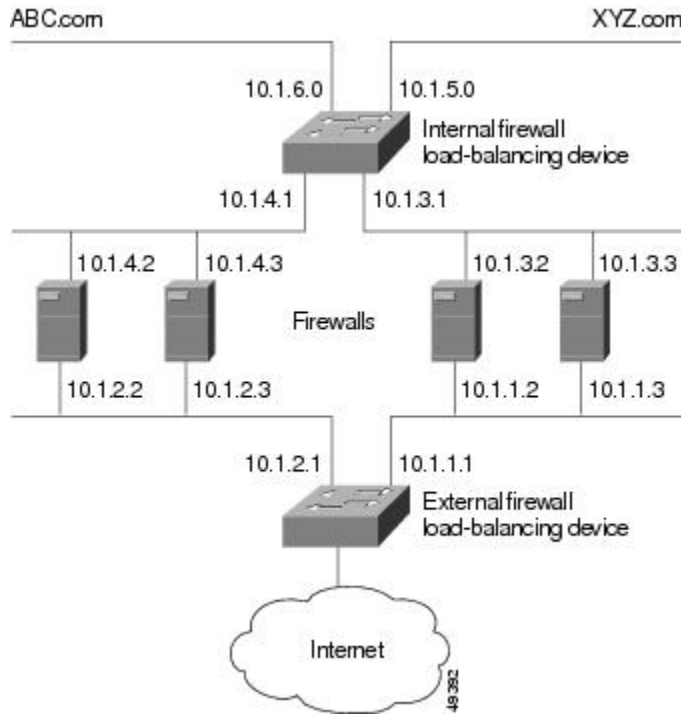
Example How to Configure IOS SLB with Multiple Firewall Farms

The figure below shows a sample IOS SLB load-balancing network with multiple firewall farms and the following components:

- Four firewalls with IP addresses as shown
- An internal firewall load-balancing device on the secure side of the firewalls
- An external firewall load-balancing device on the Internet side of the firewalls
- One firewall farm named ABCFARM, containing the two firewalls on the left.

- One firewall farm named XYZFARM, containing the two firewalls on the right.

Figure 5: IOS SLB with Multiple Firewall Farms

In the following firewall farm configuration samples:

- The internal (secure side) firewall load-balancing device is configured with firewall IP addresses 10.1.3.1 and 10.1.4.1.
- The external (Internet side) firewall load-balancing device is configured with firewall IP addresses 10.1.1.2 and 10.1.2.2.

Internal Firewall Load-Balancing Device

The following example shows the configuration for ping probes ABCPROBE and XYZPROBE and firewall farms ABCFARM and XYZFARM for the load-balancing device on the internal (secure) side of the firewalls:

```
ip slb probe ABCPROBE ping
  address 10.1.2.1
ip slb probe XYZPROBE ping
  address 10.1.1.1
ip slb firewallfarm ABCFARM
  access source 10.1.6.0 255.255.255.0
  inservice
  real 10.1.4.2
    probe ABCPROBE
    inservice
  real 10.1.4.3
    probe ABCPROBE
```

```
    inservice
ip slb firewallfarm XYZFARM
  access source 10.1.5.0 255.255.255.0
  inservice
  real 10.1.3.2
    probe XYZPROBE
    inservice
  real 10.1.3.3
    probe XYZPROBE
    inservice
```

External Firewall Load-Balancing Device

The following example shows the configuration for ping probes ABCPROBE and XYZPROBE and firewall farms ABCFARM and XYZFARM for the load-balancing device on the external (Internet) side of the firewalls:

```
ip slb probe ABCPROBE ping
  address 10.1.4.1
ip slb probe XYZPROBE ping
  address 10.1.3.1
ip slb firewallfarm ABCFARM
  access destination 10.1.6.0 255.255.255.0
  inservice
  real 10.1.2.2
    probe ABCPROBE
    inservice
  real 10.1.2.3
    probe ABCPROBE
    inservice
ip slb firewallfarm XYZFARM
  access destination 10.1.5.0 255.255.255.0
  inservice
  real 10.1.1.2
    probe XYZPROBE
    inservice
  real 10.1.1.3
    probe XYZPROBE
    inservice
```

# Example How to Configure IOS SLB with Dual Firewall Load Balancing "Sandwich"

The figure below illustrates a basic dual firewall load balancing "sandwich" configuration hosted on one IOS SLB device, including Virtual Private Network (VPN) routing and forwarding (VRF) and access interface configuration. VL105, VL106, VL107, and VL108 are VLANs.

![Note icon]

Note The client and server in this configuration are directly connected; in a more typical deployment, additional routes would be needed inside and outside the VRF.

Figure 6: IOS SLB with Dual Firewall Load Balancing "Sandwich"
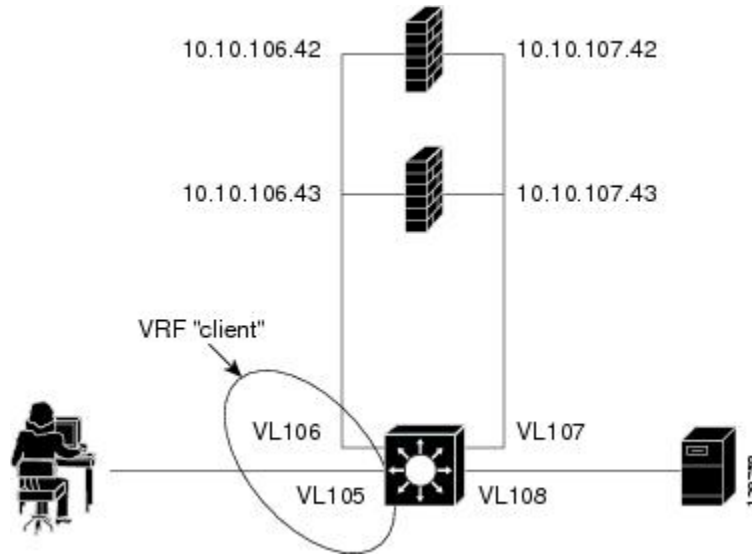
Following are the IOS SLB configuration statements for the configuration shown in the figure above:

```
ip vrf client
 rd 0:1
!
ip slb probe P642 ping
 address 10.10.106.42
 interval 120
ip slb probe P643 ping
 address 10.10.106.43
 interval 120
ip slb probe P742 ping
 address 10.10.107.42
 interval 120
ip slb probe P743 ping
 address 10.10.107.43
 interval 120
!
ip slb firewallfarm CLIENT
 access inbound Vlan105
 access outbound Vlan106
 no inservice
!
 real 10.10.106.42
  probe P642
  inservice
 real 10.10.106.43
  probe P643
  inservice
 protocol tcp
  sticky 180 source
 protocol datagram
  sticky 180 source
 predictor hash address port
!
ip slb firewallfarm SERVER
```

```
 access inbound Vlan108
 access outbound Vlan107
 inservice
!
 real 10.10.107.42
  probe P742
  inservice
 real 10.10.107.43
  probe P743
  inservice
 protocol tcp
  sticky 180 destination
 protocol datagram
  sticky 180 destination
 predictor hash address port
!
mls flow ip interface-full
!
!************************************************
!* Switchports, port channels and trunks        *
!* added to vlans 105-108 (left out for brevity) *
!************************************************
!
interface Vlan105
 ip vrf forwarding client
 ip address 10.10.105.2 255.255.255.0
!
interface Vlan106
 ip vrf forwarding client
 ip address 10.10.106.2 255.255.255.0
!
interface Vlan107
 ip address 10.10.107.2 255.255.255.0
!
interface Vlan108
 ip address 10.10.108.2 255.255.255.0
!
ip route 10.10.105.0 255.255.255.0 10.10.107.42
ip route vrf client 10.10.108.0 255.255.255.0 10.10.106.42
```

Examples How to Configure IOS SLB with Probes

This section contains the following examples, illustrating several different IOS SLB probe configurations:

-
-

Example How to Configure IOS SLB with Ping and HTTP Probes

The figure below shows a sample configuration with IOS SLB real server connections configured as part of a server farm, focusing on using ping and HTTP probes to monitor applications that are server load-balanced.

Figure 7: Sample Ping and HTTP Probe Topology
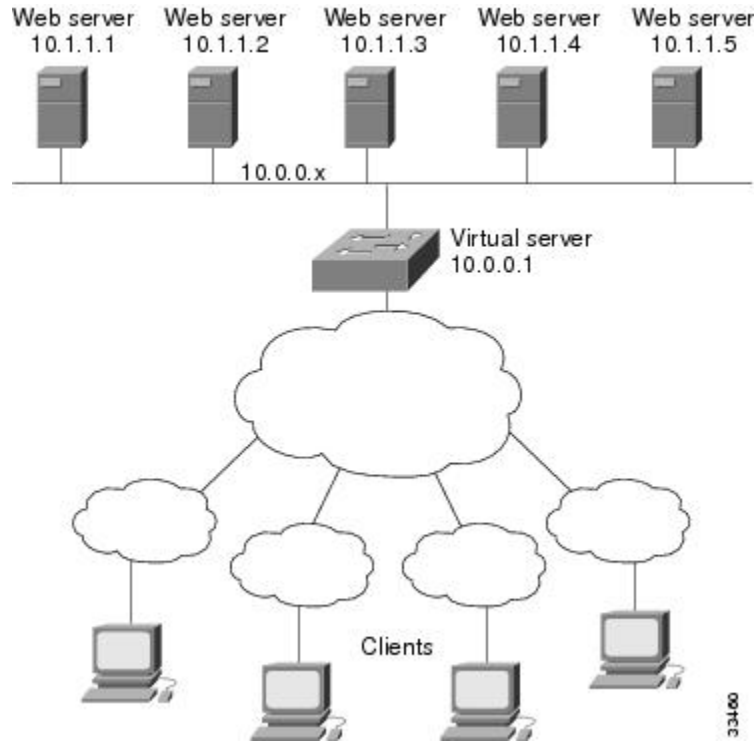
The topology shown in the figure above is a heterogeneous server farm servicing one virtual server. Following are the configuration statements for this topology, including a ping probe named PROBE1 and an HTTP probe named PROBE2:

```
! Configure ping probe PROBE1, change CLI to IOS SLB probe configuration mode
ip slb probe PROBE1 ping
! Configure probe to receive responses from IP address 13.13.13.13
  address 13.13.13.13
! Configure unacknowledged ping threshold to 16
  faildetect 16
! Configure ping probe timer interval to send every 11 seconds
  interval 11
! Configure HTTP probe PROBE2
  ip slb probe PROBE2 http
! Configure request method as POST, set URL as /probe.cgi?all
  request method post url /probe.cgi?all
! Configure header HeaderName
  header HeaderName HeaderValue
! Configure basic authentication username and password
  credentials Semisweet chips
! Exit to global configuration mode
  exit
! Enter server farm configuration mode for server farm PUBLIC
ip slb serverfarm PUBLIC
! Configure NAT server and real servers on the server farm
  nat server
  real 10.1.1.1
   inservice
  real 10.1.1.2
```

```
   inservice
 real 10.1.1.3
  inservice
 real 10.1.1.4
  inservice
 real 10.1.1.5
  inservice
! Configure ping probe on the server farm
 probe PROBE1
! Configure HTTP probe on the server farm
 probe PROBE2
 end
```

Example How to Configure IOS SLB with Routed Probe

The figure below shows a typical datacenter and IOS SLB configuration. Virtual server ACME_VSERVER is configured with two real servers (10.10.10.1 and 10.10.10.2) in server farm ACME_FARM. The user wants the real servers to fail based on the health of the backend server (10.10.10.3). To accomplish this configuration without sending health checks through the real servers, the user defines BACKEND, a routed ping probe to the backend server's IP address.

Figure 8: IOS SLB with a Routed Ping Probe
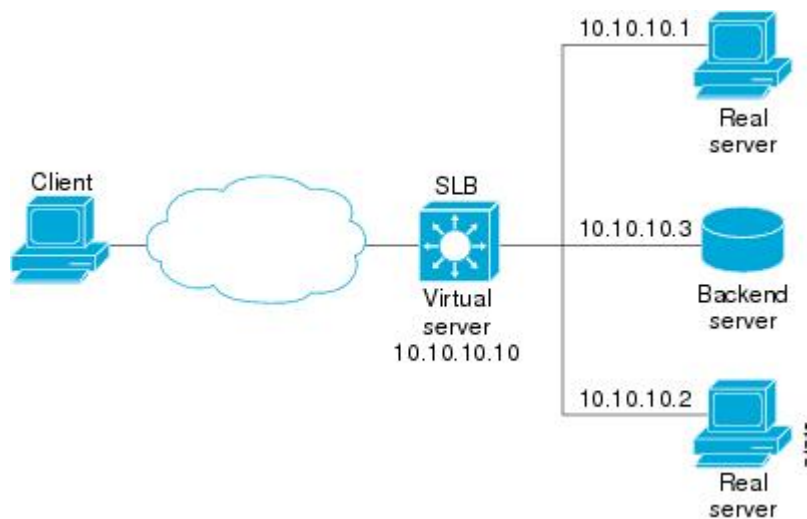
Following are the IOS SLB configuration statements for the configuration shown in the figure above:

```
ip slb probe BACKEND ping
  address 10.10.10.3 routed
ip slb serverfarm ACME_SFARM
  nat server
  probe BACKEND
  real 10.10.10.1
   inservice
  real 10.10.10.2
   inservice
ip slb vserver ACME_VSERVER
  virtual 10.10.10.10 tcp 80
  serverfarm ACME_SFARM
  inservice
```

Example How to Configure a Layer 3 Switch with IOS SLB

The figure below shows a sample configuration with IOS SLB server connections configured as part of a server farm.

Figure 9: Network Configuration for IOS SLB
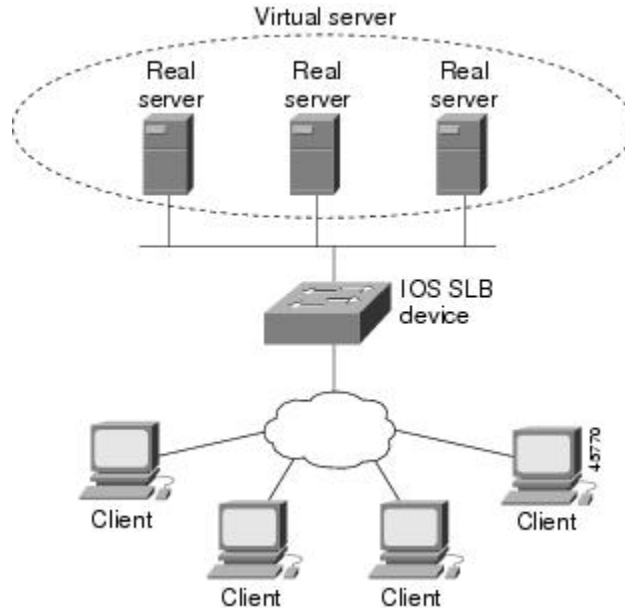
As shown in the following sample configuration, the example topology has three public web servers and two restricted web servers for privileged clients in subnet 10.4.4.0. The public web servers are weighted according to their capacity, with server 10.1.1.2 having the lowest capacity and having a connection limit imposed on it. The restricted web servers are configured as members of the same sticky group, so that HTTP connections and Secure Socket Layer (SSL) connections from the same client use the same real server.

The network configuration to provide the previously described IOS SLB functionality follows:

```
ip slb probe PROBE2 http
  request method POST url /probe.cgi?all
  header HeaderName HeaderValue
  header Authorization Basic U2VtaXN3ZWV0OmNoaXBz
!
ip slb serverfarm PUBLIC
  nat server
  predictor leastconns
! First real server
  real 10.1.1.1
    reassign 4
    faildetect numconns 16
    retry 120
    inservice
! Second real server
  real 10.1.1.2
    reassign 4
    faildetect numconns 16
    retry 120
    inservice
! Third real server
  real 10.1.1.3
```

```
                      reassign 4
                      faildetect numconns 16
                      retry 120
                      inservice
               ! Probe
                 probe PROBE2
               ! Restricted web server farm
               ip slb serverfarm RESTRICTED
                 predictor leastconns
               ! First real server
                 real 10.1.1.20
                      reassign 2
                      faildetect numconns 4
                      retry 20
                      inservice
               ! Second real server
                 real 10.1.1.21
                      reassign 2
                      faildetect numconns 4
                      retry 20
                      inservice
               !
               ! Unrestricted web virtual server
               ip slb vserver PUBLIC_HTTP
                 virtual 10.0.0.1 tcp www
                 serverfarm PUBLIC
                 idle 120
                 delay 5
                 inservice
               !
               ! Restricted HTTP virtual server
               ip slb vserver RESTRICTED_HTTP
                 virtual 10.0.0.1 tcp www
                 serverfarm RESTRICTED
                 client 10.4.4.0 255.255.255.0
                 sticky 60 group 1
                 idle 120
                 delay 5
                 inservice
               !
               ! Restricted SSL virtual server
               ip slb vserver RESTRICTED_SSL
                 virtual 10.0.0.1 tcp https
                 serverfarm RESTRICTED
                 client 10.4.4.0 255.255.255.0
                 sticky 60 group 1
                 idle 120
                 delay 5
                 inservice
               !
               interface GigabitEthernet1/1
                 switchport
                 switchport access vlan 3
                 switchport mode access
                 no ip address
               !
               interface FastEthernet2/1
                 switchport
                 switchport access vlan 2
                 switchport mode access
                 no ip address
               !
               interface FastEthernet2/2
                 switchport
                 switchport access vlan 2
                 switchport mode access
                 no ip address
               !
               interface FastEthernet2/3
                 switchport
                 switchport access vlan 2
                 switchport mode access
                 no ip address
```

```
!
interface Vlan2
  ip address 10.1.1.100 255.255.255.0
!
interface Vlan3
  ip address 40.40.40.1 255.255.255.0
```

Examples How to Configure IOS SLB with NAT and Static NAT

This section contains the following examples, illustrating several different IOS SLB NAT configurations:

Example How to Configure IOS SLB with NAT

The figure below shows a sample configuration with IOS SLB real server connections configured as part of a server farm, focusing on the configuration of the NAT server and address pool of clients.

Figure 10: Sample IOS SLB NAT Topology
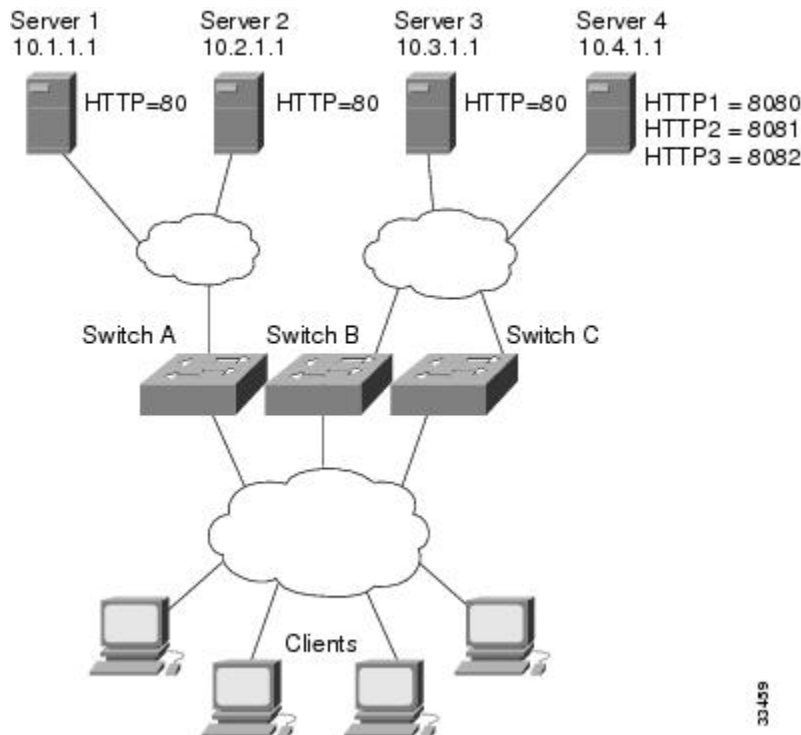
The topology in the figure above has four web servers, configured as follows:

- Servers 1, 2, and 3 are running single HTTP server applications listening on port 80.
- Server 4 has multiple HTTP server applications listening on ports 8080, 8081, and 8082.

Server 1 and Server 2 are load-balanced using Switch A, which is performing server address translation.

Server 3 and Server 4 are load-balanced using Switch B and Switch C. These two switches are performing both server and client address translation since there are multiple paths between the clients and the servers. These switches also must perform server port translation for HTTP packets to and from Server 4.

Switch A Configuration Statements

```
ip slb serverfarm FARM1
! Translate server addresses
  nat server
! Server 1 port 80
  real 10.1.1.1
    reassign 2
    faildetect numconns 4 numclients 2
    retry 20
    inservice
! Server 2 port 80
  real 10.2.1.1
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
!
ip slb vserver HTTP1
! Manage HTTP (port 80) requests
  virtual 128.1.0.1 tcp www
  serverfarm FARM1
  idle 120
  delay 5
  inservice
```

Switch B Configuration Statements

```
ip slb natpool web-clients 128.3.0.1 128.3.0.254
! NAT address pool for clients
ip slb serverfarm FARM2
! Translate server addresses
  nat server
! Translate client addresses
  nat client web-clients
! Server 3 port 80
  real 10.3.1.1
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
! Server 4 port 8080
  real 10.4.1.1 port 8080
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
! Server 4 port 8081
  real 10.4.1.1 port 8081
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
! Server 4 port 8082
  real 10.4.1.1 port 8082
    reassign 2
    faildetect numconns 4
    retry 20
```

```
      inservice
!
ip slb vserver HTTP2
! Manage HTTP (port 80) requests
  virtual 128.2.0.1 tcp www
  serverfarm FARM2
  idle 120
  delay 5
  inservice
```

Switch C Configuration Statements

```
ip slb natpool web-clients 128.5.0.1 128.5.0.254
! NAT address pool for clients
ip slb serverfarm FARM2
! Translate server addresses
  nat server
! Translate client addresses
  nat client web-clients
! Server 3 port 80
  real 10.3.1.1
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
! Server 4 port 8080
  real 10.4.1.1 port 8080
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
! Server 4 port 8081
  real 10.4.1.1 port 8081
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
! Server 4 port 8082
  real 10.4.1.1 port 8082
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
!
ip slb vserver HTTP2
! Manage HTTP (port 80) requests
  virtual 128.4.0.1 tcp www
  serverfarm FARM2
  idle 120
  delay 5
  inservice
```

Example How to Configure IOS SLB with Static NAT

The following example shows configuration statements for the following items:

- A DNS probe, PROBE4, configured to supply real server IP address 13.13.13.13 in response to domain name resolve requests.
- A server farm, DNS, that is configured to use server NAT and PROBE4.
- An all-port virtual server, 10.11.11.11, associated with server farm DNS, that performs per-packet server load balancing for UDP connections.
- A real server, 10.1.1.3, associated with server farm DNS, configured for static NAT and per-packet server load balancing.

```
ip slb probe PROBE4 dns
```

```
 lookup 13.13.13.13
!
ip slb serverfarm DNS
 nat server
 probe PROBE4
 real 10.1.1.3
  inservice
!
ip slb vserver DNS
 virtual 10.11.11.11 UDP 0 service per-packet
 serverfarm DNS
!
ip slb static nat 10.11.11.11 per-packet
 real 10.1.1.3
```

Examples How to Configure IOS SLB with Redundancy

This section contains the following examples, illustrating several different IOS SLB configurations with redundancy:

Examples How to Configure IOS SLB with Stateless Backup

There are several different ways in which you can configure IOS SLB stateless backup. The differences between the configurations depend on the networking capabilities of your load balancing devices, and on the capabilities of the distribution devices that direct client traffic to those load balancing devices.

- If a load balancing device is *not* capable of both Layer 2 switching and VLAN trunking, you must connect it and its real servers to a Layer 2 switch. This configuration is required in order to use HSRP on the server-side VLANs.
- If a distribution device is capable of Layer 3 switching, it can use route redistribution to direct flows to the active load balancing device.
- If a distribution device is capable of Layer 2 switching, it can use client-side HSRP on the load balancing device to direct flows to the active load balancing device.
- While HSRP offers faster failover times, routing converges quickly enough for most configurations. If you use both client-side and server-side HSRP on the load balancing devices, you must use HSRP interface tracking and priorities to synchronize the client-side and server-side HSRP groups.

This section contains the following examples, illustrating several different IOS SLB stateless backup configurations:

Note Stateful backup is omitted from these examples in the interest of simplicity. To see an example that uses stateful backup, see the "Example: How to Configure IOS SLB with Stateful Backup" section.
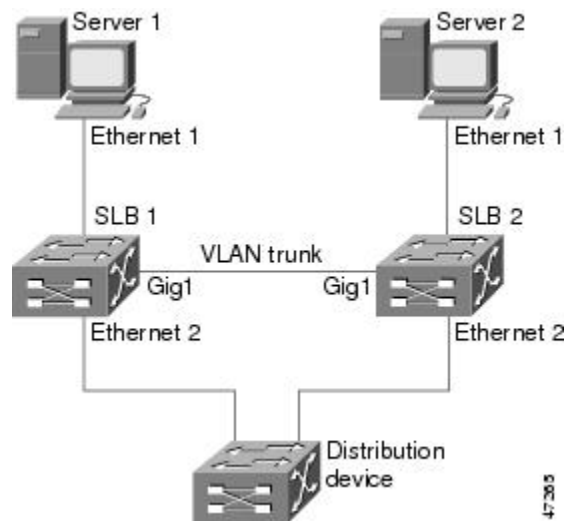
Example How to Configure Dynamic Routing and Trunking

The figure below shows a sample IOS SLB stateless backup configuration with the following characteristics:

- The IP address for real server 1 is 10.10.1.3, and for real server 2 is 10.10.1.4, routed to clients through 10.10.1.100.
- The IP address for the virtual server is 10.10.14.1.
- The IP address for VLAN 1 is 10.10.1.0, with a subnet mask of 255.255.255.0.
- The IP address for Subnet 2 is 10.10.2.0, with a subnet mask of 255.255.255.0.
- The IP address for Subnet 3 is 10.10.3.0, with a subnet mask of 255.255.255.0.
- The distribution device uses EIGRP to learn the route to 10.10.14.1 through either 10.10.2.1 or 10.10.3.1, depending on which IOS SLB is active.

Figure 11: Stateless Backup with Layer 3 and Trunking

SLB 1 Configuration Statements

```
ip slb serverfarm SF1
  real 10.10.1.3
    reassign 2
    faildetect numconns 4 numclients 2
    retry 20
    inservice
  real 10.10.1.4
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
ip slb vserver VS1
  virtual 10.10.14.1 tcp www
  serverfarm SF1
  idle 120
  delay 5
  inservice standby SERVER
```

```
!
interface Ethernet1
  switchport
  switchport vlan 1
interface Ethernet2
  ip address 10.10.2.1 255.255.255.0
interface vlan 1
  ip address 10.10.1.1 255.255.255.0
  standby ip 10.10.1.100
  standby priority 10 preempt delay sync 20
  standby name SERVER
  standby track Ethernet2
  standby timers 1 3
router eigrp 666
  redistribute static
  network 10.0.0.0
```

SLB 2 Configuration Statements

```
ip slb serverfarm SF1
  real 10.10.1.3
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
  real 10.10.1.4
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
ip slb vserver VS1
  virtual 10.10.14.1 tcp www
  serverfarm SF1
  idle 120
  delay 5
  inservice standby SERVER
!
interface GigabitEthernet1
  no ip address
  switchport
  switchport trunk encapsulation isl
interface Ethernet1
  switchport
  switchport vlan 1
interface Ethernet2
  ip address 10.10.3.1 255.255.255.0
interface vlan 1
  ip address 10.10.1.2 255.255.255.0
  standby ip 10.10.1.100
  standby priority 5 preempt delay sync 20
  standby name SERVER
  standby track Ethernet2
  standby timers 1 3
router eigrp 666
  redistribute static
  network 10.0.0.0
```
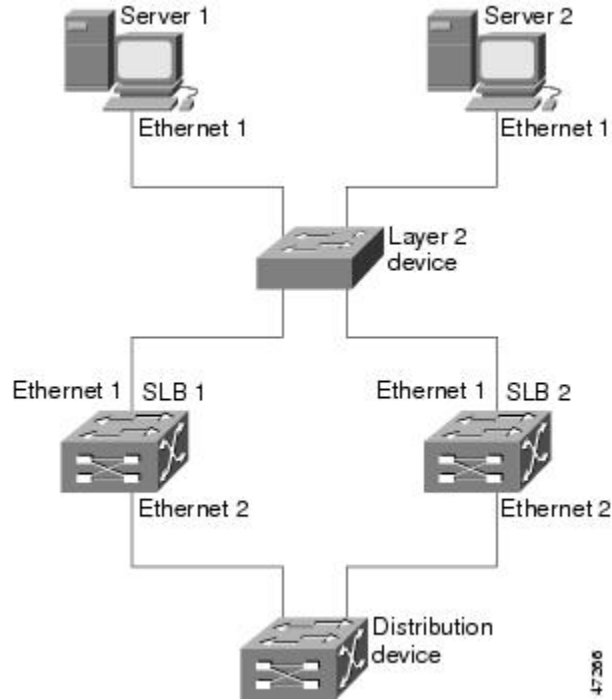
Example How to Configure Dynamic Routing and No Trunking

The figure below shows a sample IOS SLB stateless backup configuration with the following characteristics:

- The IP address for real server 1 is 10.10.1.3, and for real server 2 is 10.10.1.4, routed to clients through 10.10.1.100.
- The IP address for the virtual server is 10.10.14.1.
- The IP address for Subnet 2 is 10.10.2.0, with a subnet mask of 255.255.255.0.
- The IP address for Subnet 3 is 10.10.3.0, with a subnet mask of 255.255.255.0.

- The distribution device uses EIGRP to learn the route to 10.10.14.1 through either 10.10.2.2 or 10.10.3.2, depending on which IOS SLB is active.

Figure 12: Stateless Backup with Layer 3 and No Trunking

SLB 1 Configuration Statements

```
ip slb serverfarm SF1
  real 10.10.1.3
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
  real 10.10.1.4
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
ip slb vserver VS1
  virtual 10.10.14.1 tcp www
  serverfarm SF1
  idle 120
  delay 5
  inservice standby SERVER
!
interface Ethernet1
  ip address 10.10.1.1 255.255.255.0
  standby ip 10.10.1.100
  standby priority 10 preempt delay sync 20
  standby name SERVER
  standby track Ethernet2
  standby timers 1 3
```

```
interface Ethernet2
  ip address 10.10.2.1 255.255.255.0
router eigrp 666
  redistribute static
  network 10.0.0.0
```
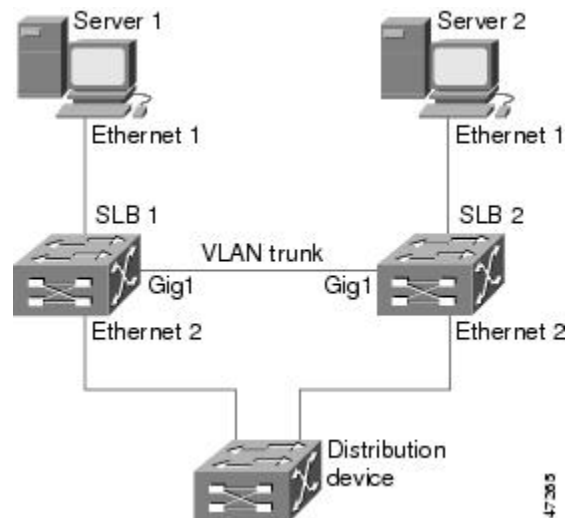
SLB 2 Configuration Statements

```
ip slb serverfarm SF1
  real 10.10.1.3
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
  real 10.10.1.4
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
ip slb vserver VS1
  virtual 10.10.14.1 tcp www
  serverfarm SF1
  idle 120
  delay 5
  inservice standby SERVER
!
interface Ethernet1
  ip address 10.10.1.2 255.255.255.0
  standby ip 10.10.1.100
  standby priority 5 preempt delay sync 20
  standby name SERVER
  standby track Ethernet2
  standby timers 1 3
interface Ethernet2
  ip address 10.10.3.1 255.255.255.0
router eigrp 666
  redistribute static
  network 10.0.0.0
```

Example How to Configure Static Routing and Trunking

The figure below shows a sample IOS SLB stateless backup configuration with the following characteristics:

- The IP address for real server 1 is 10.10.1.3, and for real server 2 is 10.10.1.4, routed to clients through 10.10.1.100.
- The IP address for the virtual server is 10.10.14.1.
- The IP address for VLAN 1 is 10.10.1.0, with a subnet mask of 255.255.255.0.
- The IP address for Subnet 2 is 10.10.2.0, with a subnet mask of 255.255.255.0.
- The IP address for Subnet 3 is 10.10.3.0, with a subnet mask of 255.255.255.0.

- The configuration uses static routing to the HSRP route on the distribution device.

Figure 13: Stateless Backup with Layer 2 and Trunking

SLB 1 Configuration Statements

```
ip slb serverfarm SF1
  real 10.10.1.3
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
  real 10.10.1.4
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
ip slb vserver VS1
  virtual 10.10.14.1 tcp www
  serverfarm SF1
  idle 120
  delay 5
  inservice standby SERVER
!
interface Ethernet1
  switchport
  switchport vlan 1
interface Ethernet2
  ip address 10.10.2.1 255.255.255.0
  standby ip 10.10.2.100
  standby priority 10 preempt delay sync 20
  standby track vlan1
  standby timers 1 3
interface vlan 1
  ip address 10.10.1.1 255.255.255.0
  standby ip 10.10.1.100
  standby priority 10 preempt delay sync 20
  standby name SERVER
```

```
                              standby track Ethernet2
                              standby timers 1 3
```

SLB 2 Configuration Statements

```
              ip slb serverfarm SF1
                real 10.10.1.3
                  reassign 2
                  faildetect numconns 4
                  retry 20
                  inservice
                real 10.10.1.4
                  reassign 2
                  faildetect numconns 4
                  retry 20
                  inservice
              ip slb vserver VS1
                virtual 10.10.14.1 tcp www
                serverfarm SF1
                idle 120
                delay 5
                inservice standby SERVER
              !
              interface GigabitEthernet1
                no ip address
                switchport
                switchport trunk encapsulation isl
              interface Ethernet1
                switchport
                switchport vlan 1
              interface Ethernet2
                ip address 10.10.2.2 255.255.255.0
                standby ip 10.10.2.100
                standby priority 5 preempt delay sync 20
                standby track vlan 1
                standby timers 1 3
              interface vlan 1
                ip address 10.10.1.2 255.255.255.0
                standby ip 10.10.1.100
                standby priority 5 preempt delay sync 20
                standby name SERVER
                standby track Ethernet2
                standby timers 1 3
```

Distribution Device Configuration Statements

```
              interface Ethernet1
                switchport
                switchport distribution vlan 2
              interface Ethernet2
                switchport
                switchport distribution vlan 2
              interface vlan2
                ip address 10.10.2.3 255.255.255.0
                no shut
              ip route 10.10.14.1 255.255.255.255 10.10.2.100
```
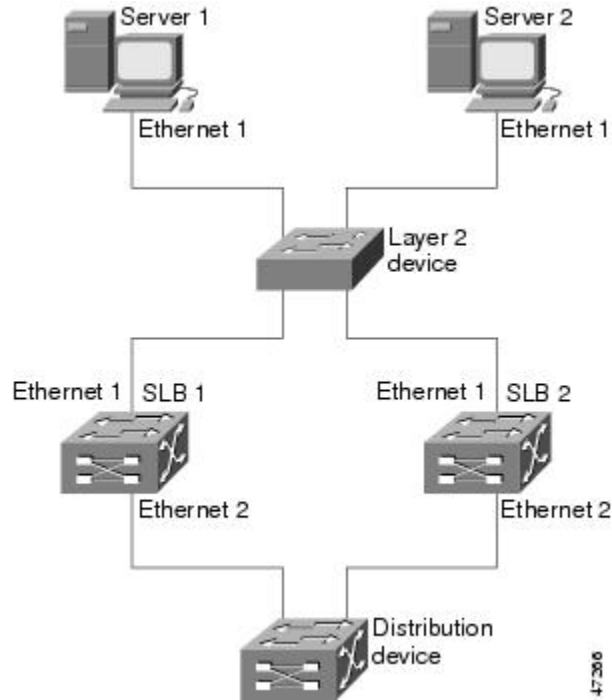
Example How to Configure Static Routing and No Trunking

The figure below shows a sample IOS SLB stateless backup configuration with the following characteristics:

- The IP address for real server 1 is 10.10.1.3, and for real server 2 is 10.10.1.4, routed to clients through 10.10.1.100.
- The IP address for the virtual server is 10.10.14.1.
- The IP address for Subnet 2 is 10.10.2.0, with a subnet mask of 255.255.255.0.

- The IP address for Subnet 3 is 10.10.3.0, with a subnet mask of 255.255.255.0.
- The configuration uses static routing to the HSRP route on the distribution device.

Figure 14: Stateless Backup with Layer 2 and No Trunking

SLB 1 Configuration Statements

```
ip slb serverfarm SF1
  real 10.10.1.3
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
  real 10.10.1.4
    reassign 2
    faildetect numconns 4
    retry 20
    inservice
ip slb vserver VS1
  virtual 10.10.14.1 tcp www
  serverfarm SF1
  idle 120
  delay 5
  inservice standby SERVER
!
interface Ethernet1
  ip address 10.10.1.1 255.255.255.0
  standby ip 10.10.1.100
  standby priority 10 preempt delay sync 20
  standby name SERVER
  standby track Ethernet2
```

```
                     standby timers 1 3
                interface Ethernet2
                  ip address 10.10.2.1 255.255.255.0
                  standby ip 10.10.2.100
                  standby priority 10 preempt delay sync 20
                  standby track Ethernet1
                  standby timers 1 3
```

SLB 2 Configuration Statements

```
                ip slb serverfarm SF1
                  real 10.10.1.3
                    reassign 2
                    faildetect numconns 4
                    retry 20
                    inservice
                  real 10.10.1.4
                    reassign 2
                    faildetect numconns 4
                    retry 20
                    inservice
                ip slb vserver VS1
                  virtual 10.10.14.1 tcp www
                  serverfarm SF1
                  idle 120
                  delay 5
                  inservice standby SERVER
                !
                interface Ethernet1
                  ip address 10.10.1.2 255.255.255.0
                  standby ip 10.10.1.100
                  standby priority 5 preempt delay sync 20
                  standby name SERVER
                  standby track Ethernet2
                  standby timers 1 3
                !
                interface Ethernet2
                  ip address 10.10.2.2 255.255.255.0
                  standby ip 10.10.2.100
                  standby priority 5 preempt delay sync 20
                  standby track Ethernet1
                  standby timers 1 3
```

Distribution Device Configuration Statements

```
                interface Ethernet1
                  switchport
                  switchport distribution vlan 2
                interface Ethernet2
                  switchport
                  switchport distribution vlan 2
                interface vlan2
                  ip address 10.10.2.3 255.255.255.0
                  no shut
                ip route 10.10.14.1 255.255.255.255 10.10.2.100
```

Example How to Configure IOS SLB with Stateful Backup

This sample configuration focuses on the IOS SLB real server connections configured as part of a server farm, with real and virtual servers over Fast Ethernet interfaces configured with stateful backup standby connections.

The figure below is an example of a stateful backup configuration, using HSRP on both the client and server sides to manage failover. The real servers route outbound flows to 10.10.3.100, which is the HSRP address on the server side interfaces. The client (or access router), routes to the virtual IP address (10.10.10.12) through 10.10.2.100, HSRP address on the client side.

Notice the loopback interfaces configured on both devices for the exchange of these messages. Each IOS SLB should also be given duplicate routes to the other switch loopback address. This configuration allows replication messages to flow despite an interface failure.

Note   To allow HSRP to function properly, the **set spantree portfast** command must be configured on any Layer 2 device between the IOS SLB switches.

Figure 15: IOS SLB Stateful Environment
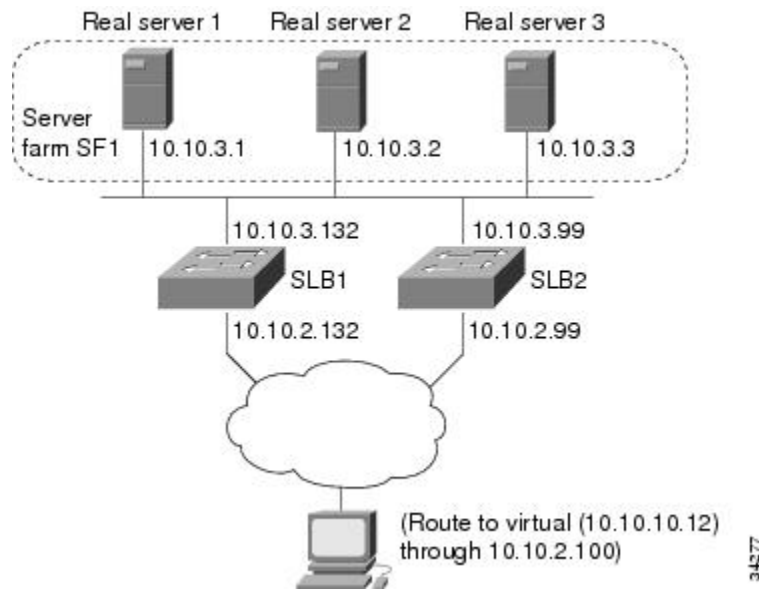
- Switch SLB1 Configuration Statements, page 135
- Switch SLB2 Configuration Statements, page 136

Switch SLB1 Configuration Statements

```
ip slb serverfarm SF1
  nat server
  real 10.10.3.1
   inservice
  real 10.10.3.2
   inservice
  real 10.10.3.3
   inservice
!
ip slb vserver VS1
  virtual 10.10.10.12 tcp telnet
  serverfarm SF1
  replicate casa 10.10.99.132 10.10.99.99 1024 password PASS
  inservice standby virt
!
interface loopback 1
  ip address 10.10.99.132 255.255.255.255
!
interface FastEthernet1
  ip address 10.10.3.132 255.255.255.0
  no ip redirects
```

```
                    no ip mroute-cache
                    standby priority 5 preempt
                    standby name out
                    standby ip 10.10.3.100
                    standby track FastEthernet2
                    standby timers 1 3
                  interface FastEthernet2
                    ip address 10.10.2.132 255.255.255.0
                    no ip redirects
                    standby priority 5
                    standby name virt
                    standby ip 10.10.2.100
                    standby timers 1 3
```
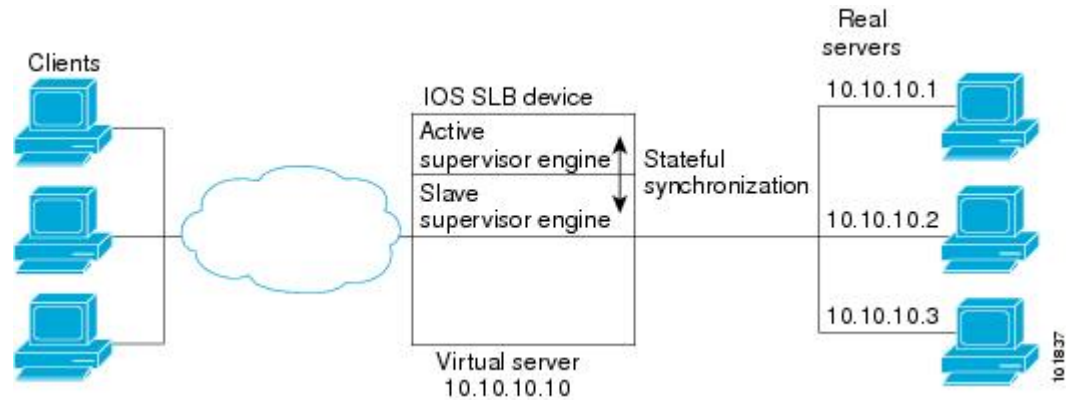
Switch SLB2 Configuration Statements

```
           ip slb serverfarm SF1
             nat server
             real 10.10.3.1
              inservice
             real 10.10.3.2
              inservice
             real 10.10.3.3
              inservice
           !
           ip slb vserver VS1
             virtual 10.10.10.12 tcp telnet
             serverfarm SF1
             replicate casa 10.10.99.99 10.10.99.132 1024 password PASS
             inservice standby virt
           !
           interface loopback 1
             ip address 10.10.99.99 255.255.255.255
           !
           interface FastEthernet2
             ip address 10.10.2.99 255.255.255.0
             no ip redirects
             no ip route-cache
             no ip mroute-cache
             standby priority 10 preempt delay sync 20
             standby name virt
             standby ip 10.10.2.100
             standby track FastEthernet3
             standby timers 1 3
           !
           interface FastEthernet3
             ip address 10.10.3.99 255.255.255.0
             no ip redirects
             no ip route-cache
             no ip mroute-cache
             standby priority 10 preempt delay 20
             standby name out
             standby ip 10.10.3.100
             standby track FastEthernet2
             standby timers 1 3
```

Example How to Configure IOS SLB with Stateful Backup of Redundant Route Processors

In the figure below, the IOS SLB device includes two Supervisor engines configured for stateful backup. If the active Supervisor engine fails, the backup Supervisor engine takes over through RPR+ with IOS SLB synchronization information already populated. IOS SLB replicates state information for virtual server

ACME_VSERVER (10.10.10.10) from the active Supervisor engine to the backup every 20 seconds. The real servers (10.10.10.1, 10.10.10.2, and 10.10.10.3) are configured in server farm ACME_SFARM.

Figure 16: IOS SLB with Redundant Route Processors

Following are the IOS SLB configuration statements for the configuration shown in the figure above:

```
ip slb replicate slave rate 300
ip slb serverfarm ACME_SFARM
  nat server
  real 10.10.10.1
   inservice
  real 10.10.10.2
   inservice
  real 10.10.10.3
   inservice
ip slb vserver ACME_VSERVER
 virtual 10.10.10.10 tcp 80
 replicate interval 20
 replicate slave
 serverfarm ACME_SFARM
 inservice
```

Example How to Configure IOS SLB with Active Standby

The figure below shows an IOS SLB network configured for active standby, with two IOS SLB devices load-balancing the same virtual IP address while backing up each other. If either device fails, the other takes over its load through normal HSRP failover and IOS SLB stateless redundancy.

Figure 17: IOS SLB Active Standby
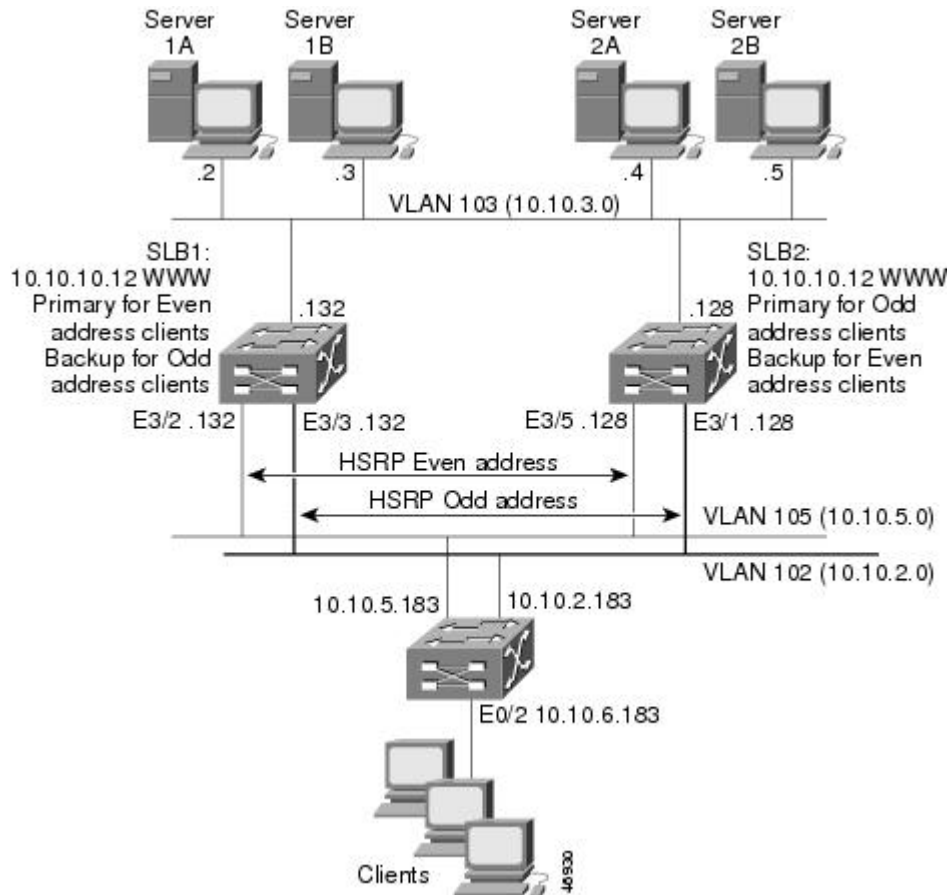
The sample network configuration in the figure above has the following characteristics:

- SLB 1 balances servers 1A and 1B and SLB 2 balances 2A and 2B.
- One virtual IP address (10.10.10.12 for web) is supported across the two IOS SLB devices.
- Client traffic is divided in an access router, sending clients with even IP addresses to HSRP1 (10.10.5.100) and clients with odd IP addresses to HSRP2 (10.10.2.100). SLB 1 is configured as primary for clients with odd IP addresses, and SLB 2 is primary for clients with even IP addresses.
- The IOS SLB devices balance the traffic to disjoint sets of real servers. (If client NAT was used in this example, this characteristic would not be a requirement).
- Each set of real servers has a default gateway configured to its IOS SLB device.
- The HSRP address on VLAN 105 is 10.10.5.100. The HSRP address on VLAN 102 is 10.10.2.100.

SLB 1 Configuration Statements

```
ip slb serverfarm EVEN
 nat server
 real 10.10.3.2
  reassign 2
  faildetect numconns 4 numclients 2
  retry 20
  inservice
 real 10.10.3.3
  reassign 2
  faildetect numconns 4
  retry 20
  inservice
!
ip slb serverfarm ODD
 nat server
 real 10.10.3.2
  reassign 2
  faildetect numconns 4
  retry 20
  inservice
 real 10.10.3.3
  reassign 2
  faildetect numconns 4
  retry 20
  inservice
!-----Same EVEN virtual server as in SLB 2
ip slb vserver EVEN
 virtual 10.10.10.12 tcp www
 serverfarm EVEN
 client 0.0.0.0 0.0.0.1
 idle 120
 delay 5
!-----See standby name in Ethernet 3/3 below
 inservice standby STANDBY_EVEN
!-----Same ODD virtual server as in SLB 2
ip slb vserver ODD
 virtual 10.10.10.12 tcp www
 serverfarm ODD
 client 0.0.0.1 0.0.0.1
 idle 120
 delay 5
!-----See standby name in Ethernet 3/2 below
 inservice standby STANDBY_ODD
!
interface Ethernet3/2
 ip address 10.10.5.132 255.255.255.0
 standby priority 20 preempt delay sync 20
!-----See standby name in SLB 2, Ethernet 3/5
 standby name STANDBY_ODD
 standby ip 10.10.5.100
 standby track Ethernet3/3
 standby timers 1 3
!
interface Ethernet3/3
 ip address 10.10.2.132 255.255.255.0
 standby priority 10
!-----See standby name in SLB 2, Ethernet 3/1
 standby name STANDBY_EVEN
 standby ip 10.10.2.100
 standby track Ethernet3/2
 standby timers 1 3
```

SLB 2 Configuration Statements

```
ip slb serverfarm EVEN
 nat server
```

```
 real 10.10.3.4
  reassign 2
  faildetect numconns 4
  retry 20
  inservice
 real 10.10.3.5
  reassign 2
  faildetect numconns 4
  retry 20
  inservice
!
ip slb serverfarm ODD
 nat server
 real 10.10.3.4
  reassign 2
  faildetect numconns 4
  retry 20
  inservice
 real 10.10.3.5
  reassign 2
  faildetect numconns 4
  retry 20
  inservice
!-----Same EVEN virtual server as in SLB 1
ip slb vserver EVEN
 virtual 10.10.10.12 tcp www
 serverfarm EVEN
 client 0.0.0.0 0.0.0.1
 idle 120
 delay 5
!-----See standby name in Ethernet 3/1 below
 inservice standby STANDBY_EVEN
!-----Same ODD virtual server as in SLB 1
ip slb vserver ODD
 virtual 10.10.10.12 tcp www
 serverfarm ODD
 client 0.0.0.1 0.0.0.1
 idle 120
 delay 5
!-----See standby name in Ethernet 3/5 below
 inservice standby STANDBY_ODD
!
interface Ethernet3/1
 ip address 10.10.2.128 255.255.255.0
 standby priority 20 preempt delay sync 20
!-----See standby name in SLB 1, Ethernet 3/3
 standby name STANDBY_EVEN
 standby ip 10.10.2.100
 standby track Ethernet3/5
 standby timers 1 3
!
interface Ethernet3/5
 ip address 10.10.5.128 255.255.255.0
 standby priority 10 preempt delay sync 20
!-----See standby name in SLB 1, Ethernet 3/2
 standby name STANDBY_ODD
 standby ip 10.10.5.100
 standby track Ethernet3/1
 standby timers 1 3
```

Access Router Configuration Statements

```
interface Ethernet0/0
 ip address 10.10.5.183 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet0/1
 ip address 10.10.2.183 255.255.255.0
 no ip directed-broadcast
```

```
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet0/2
 ip address 10.10.6.183 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 ip policy route-map virts
!
access-list 100 permit ip 0.0.0.1 255.255.255.254 host 10.10.10.12
access-list 101 permit ip 0.0.0.0 255.255.255.254 host 10.10.10.12
route-map virts permit 10
match ip address 100
set ip next-hop 10.10.5.100
!
route-map virts permit 15
match ip address 101
set ip next-hop 10.10.2.100
```

Example How to Configure IOS SLB with Redistribution of Static Routes

The figure below shows an IOS SLB network configured to distribute static routes to a virtual server's IP address. The route to the address is added to the routing table as **static** if you advertise the address when you bring the virtual server into service (using the **inservice** command). See the description of the **advertise** command in the *Cisco IOS IP Application Services Command Reference* for more details about advertising virtual server IP addresses.

Because the routing configuration varies from protocol to protocol, sample configurations for several different routing protocols are given.

Figure 18: IOS SLB Redistribution of Static Routes
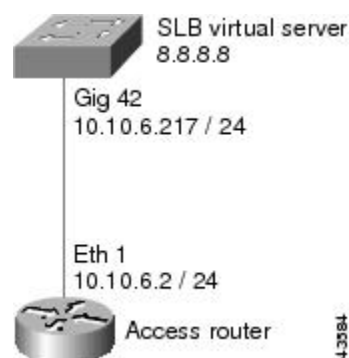
Routing Information Protocol (RIP)

Following is the RIP static route redistribution configuration for the IOS SLB switch shown in the figure above:

```
router rip
 redistribute static
 network 10.0.0.0
 network 8.0.0.0
```

Following is the RIP static route redistribution configuration for the access router that is listening for routing updates shown in the figure above:

```
router rip
 network 10.0.0.0
 network 8.0.0.0
```

Open Shortest Path First (OSPF)

Following is the OSPF static route redistribution configuration for the IOS SLB switch shown in the figure above:

```
router ospf 1
 redistribute static subnets
 network 10.10.6.217 0.0.0.0 area 0
 network 8.8.8.0 0.0.0.255 area 0
```

Following is the OSPF static route redistribution configuration for the access router that is listening for routing updates shown in the figure above:

```
router ospf 1
 network 10.10.6.2 0.0.0.0 area 0
 network 8.8.8.0 0.0.0.255 area 0
```

Interior Gateway Routing Protocol (IGRP)

Following is the IGRP static route redistribution configuration for the IOS SLB switch shown in the figure above:

```
router igrp 1
 redistribute connected
 redistribute static
 network 8.0.0.0
 network 10.0.0.0
```

Following is the IGRP static route redistribution configuration for the access router that is listening for routing updates shown in the figure above:

```
router igrp 1
 network 8.0.0.0
 network 10.0.0.0
```

Enhanced Interior Gateway Routing Protocol (Enhanced IGRP)

Following is the Enhanced IGRP static route redistribution configuration for the IOS SLB switch shown in the figure above:

```
router eigrp 666
 redistribute static
 network 10.0.0.0
 network 8.0.0.0
```

Following is the Enhanced IGRP static route redistribution configuration for the access router that is listening for routing updates shown in the figure above:

```
router eigrp 666
 network 10.0.0.0
 network 8.0.0.0
```

Example How to Configure IOS SLB with WAP and UDP Load Balancing

The figure below shows an IOS SLB network configured to balance WAP flows. In this example:

- WAP flows are balanced between WAP gateways 10.10.2.1, 10.10.2.2, and 10.10.2.3.
- The clients connect to 10.10.1.1, the IOS SLB virtual server address.
- For a given session, load-balancing decisions change if the connection idles longer than the virtual server's idle connection timer (3000 seconds in this example).

Figure 19: IOS SLB with WAP Load Balancing
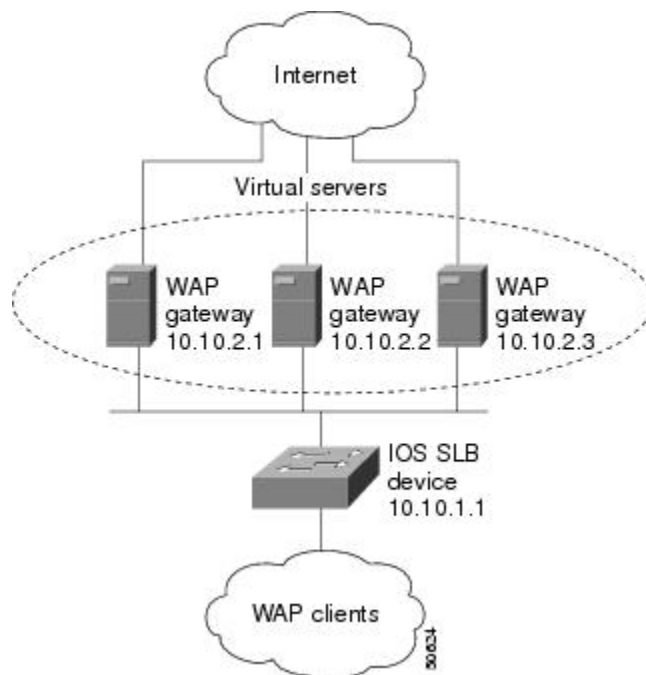
There are two ways to configure IOS SLB load balancing for WAP:

- To load-balance sessions running in connection-oriented WSP mode, define a WSP probe and use WAP load balancing. WAP load balancing requires a WAP virtual server configured on one of the WAP ports.
- To load-balance sessions running in connectionless WSP, connectionless secure WSP, and connection-oriented secure WSP modes, define a ping or WSP probe and use standard UDP load balancing with a low idle timer.

  - Example How to Balance WAP Flows on UDP Port 9201, page 144

Example How to Balance WAP Flows on UDP Port 9201

The following example shows the configuration for the IOS SLB device shown in the figure above, which balances WAP flows on UDP port 9201 (WSP/WTP/UDP):

```
ip slb probe PROBE3 wsp
  url http://localhost/test.txt
!
ip slb serverfarm WAPFARM
  nat server
  real 10.10.2.1
  inservice
  real 10.10.2.2
  inservice
  real 10.10.2.3
  inservice
  probe PROBE3
!
ip slb vserver VSERVER
  virtual 10.10.1.1 udp 9201
  serverfarm WAPFARM
  idle 3000
  inservice
```

Example How to Balance WAP Flows on UDP Port 9203

The following example shows the configuration for the IOS SLB device shown in the figure above, which balances WAP flows on UDP port 9203 (WSP/WTP/WTLS/UDP):

```
ip slb probe PROBE1 ping
!
ip slb serverfarm WAPFARM
  nat server
  real 10.10.2.1
  inservice
  real 10.10.2.2
  inservice
  real 10.10.2.3
  inservice
  probe PROBE1
!
ip slb vserver VSERVER
  virtual 10.10.1.1 udp 9203
  serverfarm WAPFARM
  idle 3000
  inservice
```

Examples How to Configure IOS SLB with Route Health Injection

This section contains the following examples, illustrating several different IOS SLB route health injection configurations:

Example How to Configure Two Distributed Sites with One Web Server Each

The figure below shows an IOS SLB network configured with route health injection with the following characteristics:

- Both IOS SLB devices are configured with the same virtual IP address.
- Each IOS SLB device has a server farm containing only the locally attached web server as a real server.
- The path to SLB A has the lower weight.

Figure 20: Two Distributed Sites with One Web Server Each
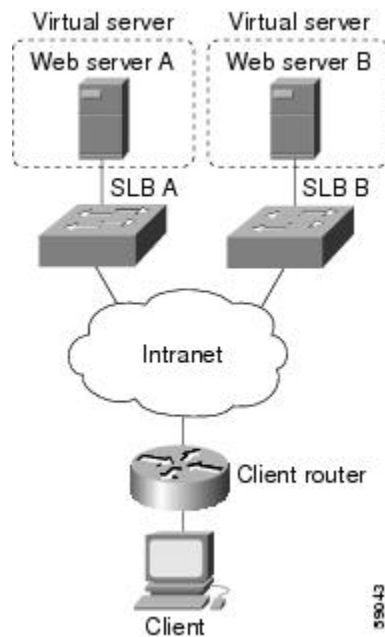
When both web servers in the figure above are operational, the client router receives the host route from both IOS SLB devices.

If Web Server A fails, the virtual server for the virtual IP address on SLB A enters FAILED state and stops advertising the host route for the virtual IP address. The client router then begins using the route to SLB B.

When Web Server A is again available, the virtual server again advertises the host route for the virtual IP address, and the client router begins using SLB A.

Example How to Configure Two Distributed Sites with Two Web Servers Each

The figure below shows an IOS SLB network configured with route health injection with the following characteristics:

- Both IOS SLB devices are configured with the same virtual IP address.
- Each IOS SLB device has a server farm containing two locally attached web servers as real servers.

- The path to SLB A has the lower weight.

Figure 21: Two Distributed Sites with Two Web Servers Each
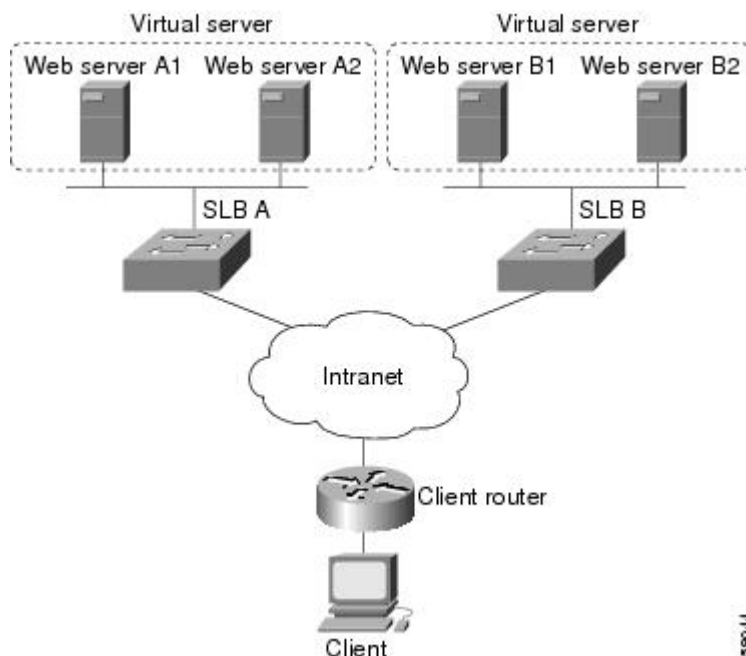
When all web servers in the figure above are operational, the client router receives the host route from both IOS SLB devices.

If one web server in either server farm fails, the route continues to be advertised by the given IOS SLB device.

If both Web Server A1 and Web Server A2 fail, the virtual server for the virtual IP address on SLB A enters FAILED state and stops advertising the host route for the virtual IP address. The client router then begins using the route to SLB B.

When either Web Server A1 or Web Server A2 is again available, the virtual server again advertises the host route for the virtual IP address, and the client router begins using SLB A.

Example How to Configure Two Distributed Sites with One Web Server and a Backup IOS SLB Switch Each

The figure below shows an IOS SLB network configured with route health injection with the following characteristics:

- Both IOS SLB devices are configured with the same virtual IP address.
- Each IOS SLB device has a server farm containing only the locally attached web server as a real server.
- Each site has a primary IOS SLB device and a backup IOS SLB device.

- The path to SLB A has the lower weight.

Figure 22: Two Distributed Sites with One Web Server and a Backup IOS SLB Switch Each
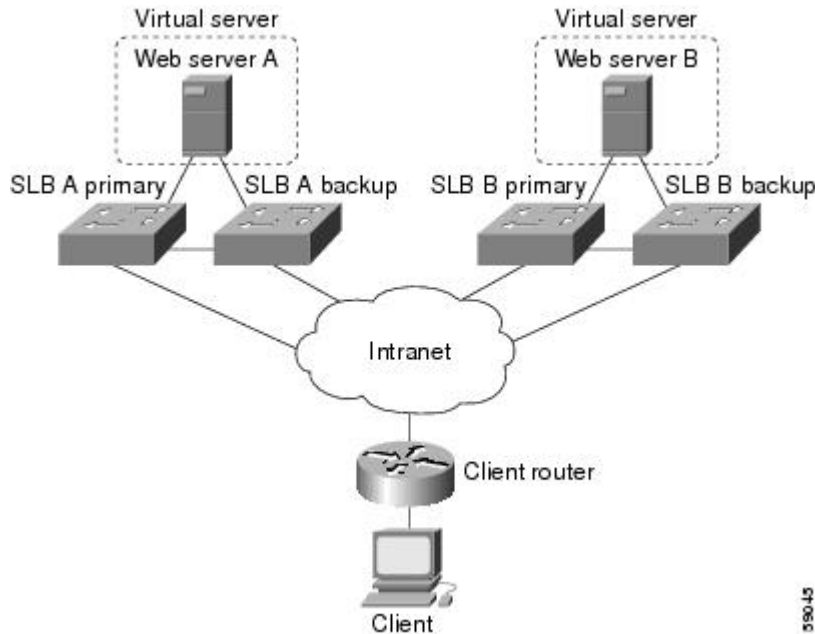
When both web servers in the figure above are operational, the client router receives the host route from both SLB A Primary and SLB B Primary.

If SLB A Primary fails, SLB A Backup begins advertising the host route to the virtual IP address. If SLB A Backup also fails, the virtual server for the virtual IP address on SLB A Primary and SLB A Backup enters FAILED state and stops advertising the host route for the virtual IP address. The client router then begins using the route to SLB B Primary (or to SLB B Backup, if SLB B Primary is not available).

When either SLB A Primary or SLB A Backup is again available, the virtual server again advertises the host route for the virtual IP address, and the client router begins using SLB A Primary or SLB A Backup.

Examples How to Configure IOS SLB with GPRS Load Balancing

This section contains the following examples, illustrating several different IOS SLB configurations with redundancy:

Example How to Configure IOS SLB with GPRS Load Balancing Without GTP Cause Code Inspection

The figure below shows a typical GPRS load-balancing configuration *without* GTP cause code inspection enabled. In this configuration:

- IOS SLB can balance GPRS flows across multiple real GGSNs. The SGSN "sees" the real GGSNs as one virtual GGSN. This configuration increases the flow-handling capability of the real GGSNs and increases the reliability and availability.
- The virtual template 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 23: IOS SLB with GPRS Load Balancing
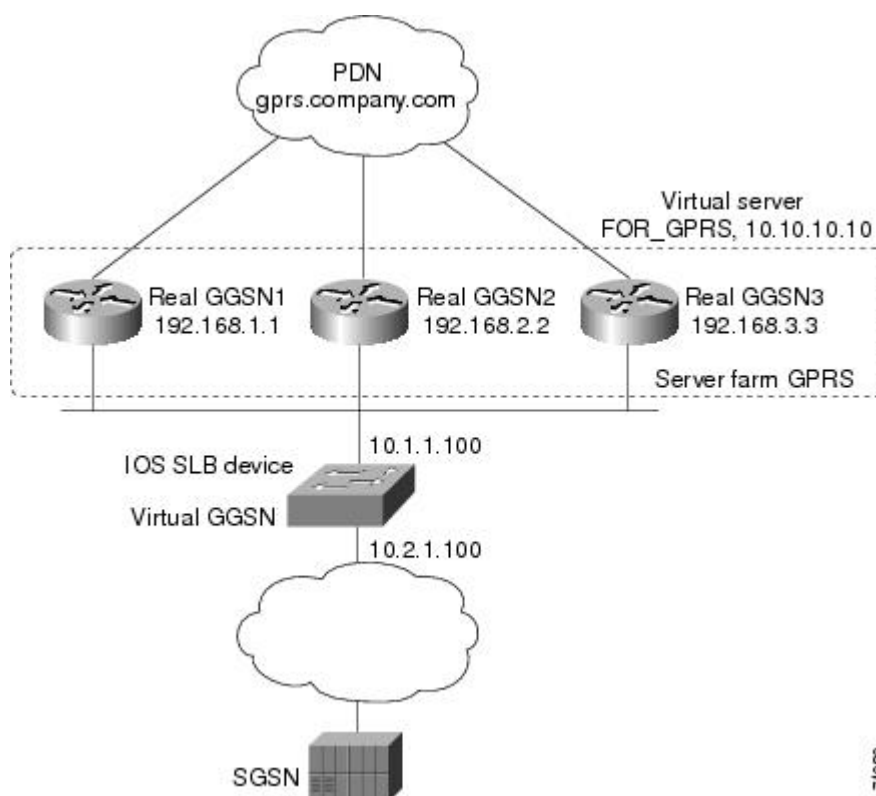
Following are the configuration statements for the configuration shown in the figure above:

For more detailed GGSN configuration examples, refer to 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 Password1 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 Password1 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 Password1 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 Password1 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
```

Example How to Configure IOS SLB with GPRS Load Balancing and NAT

The following example uses the same basic configuration as in the "Example: How to Configure IOS SLB with GPRS Load Balancing Without GTP Cause Code Inspection", including the network shown in the figure above, but with the addition of NAT:

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

- IOS SLB Configuration Statements, page 152
- GGSN1 Configuration Statements, page 152

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 Password1 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 Password1 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 Password1 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 Password1 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
```

Example How to Configure IOS SLB with GPRS Load Balancing NAT and GTP Cause Code Inspection

The following example uses the same basic configuration as in the "Example: How to Configure IOS SLB with GPRS Load Balancing and NAT" section, including the network shown in the figure above, but with the GTP cause code inspection enabled. In this configuration:

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

Following are the configuration statements for the configuration shown in the figure above, with the addition of NAT and GTP cause code inspection support:

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

-

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 Password1 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
```

Example How to Configure IOS SLB with GPRS Load Balancing Maps

The following sample configuration enables IOS SLB to support multiple server farms behind a GPRS load balancing virtual server, using access point names (APNs) to select server farms. Server farm **farm6** is configured without an associated map, and therefore acts as a default server farm. If IOS SLB cannot match any of the other server farm maps, and a default server farm is configured, IOS SLB sends the GPRS request to the default serverfarm.

```
ip slb map 1 gtp
 apn cisco*
ip slb map 4 gtp
 apn abc.microsoft.com
```

```
 apn xyz.intel.com
ip slb map 5 gtp
 apn yahoo.com
!
ip slb serverfarm farm1
 real 10.0.0.1
 inservice
 real 10.0.0.2
 inservice
ip slb serverfarm farm2
 real 10.0.0.3
 inservice
 real 10.0.0.4
 inservice
ip slb serverfarm farm3
 real 10.0.0.5
 inservice
 real 10.0.0.6
 inservice
ip slb serverfarm farm4
 real 10.0.0.7
 inservice
 real 10.0.0.8
 inservice
ip slb serverfarm farm5
 real 10.0.0.9
 inservice
 real 10.0.0.10
 inservice
ip slb serverfarm farm6
 real 10.0.0.11
 inservice
!
ip slb map 1 gtp
 apn cisco*
ip slb map 4 gtp
 apn abc.microsoft.com
 apn xyz.intel.com
ip slb map 5 gtp
 apn yahoo.com
!
ip slb vserver GGSN_SERVER
 virtual 10.10.10.10 udp 0 service gtp
 serverfarm farm1 backup farm2 map 1 priority 3
 serverfarm farm4 map 4 priority 1
 serverfarm farm5 map 5 priority 4
 serverfarm farm6
 inservice
```

Example How to Configure IOS SLB with Dual-Stack Addresses for GTP Load Balancing

The following sample configuration enables IOS SLB to support dual-stack addresses for GTP load balancing.

```
ip slb serverfarm SF1
 real 172.16.88.5
  weight 1
  inservice
!
ip slb serverfarm SF2
 real 172.16.88.6
  weight 1
  inservice
!
ip slb serverfarm SF3
 real 172.16.88.7 ipv6 2342:2342:2343:FF04:2388:BB03:3329:8612
  weight 1
  inservice
```

```
!
ip slb serverfarm SF4
 real 172.16.88.8 ipv6 2342:2342:2343:FF04:2388:BB03:3423:8912
  weight 1
  inservice
!
ip slb vserver VS2
 virtual 4.3.2.1 ipv6 2342:2342:2343:FF04:2341:AA03:2323:8912 udp 0 service gtp
 serverfarm sf1 backup sf2 ipv6-primary sf3 ipv6-backup sf4
 idle gtp request 90
 idle gtp imsi 10000000
 sticky gtp imsi group 1
 gtp notification cac 3
 inservice
```

Example How to Configure IOS SLB with VPN Server Load Balancing

The figure below shows a typical VPN server load-balancing configuration. In this configuration:

- VPN flows are balanced between real servers 20.20.20.10 and 20.20.20.20.
- Clients connect to 10.10.1.1, the IOS SLB virtual server address.
- There is a sticky connection between the ESP virtual server and the UDP virtual server.
- The cryptographic key exchange occurs through IKE (ISAKMP; port 500).

Figure 24: IOS SLB with VPN Server Load Balancing
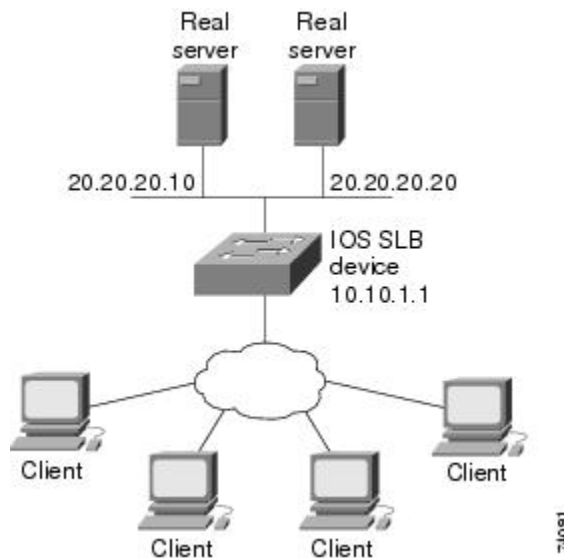
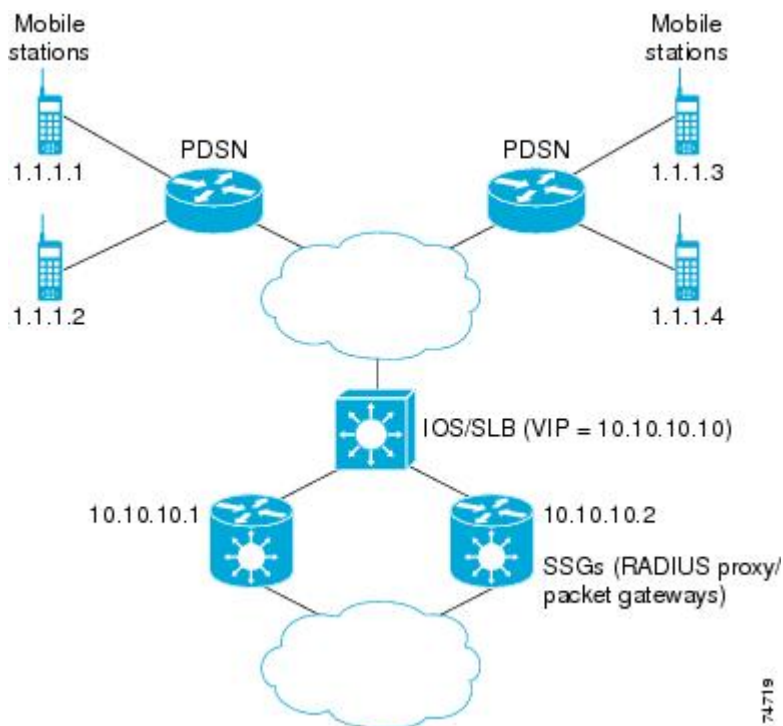Following are the IOS SLB configuration statements for the configuration shown in the figure above:

```
ip slb serverfarm VPN
 nat server
 real 20.20.20.10
  inservice
 real 20.20.20.20
  inservice
 failaction purge
!
ip slb vserver ESP
```

```
 virtual 10.10.1.1 ESP
 serverfarm VPN
 sticky 3600 group 69
 inservice
!
ip slb vserver UDP
 virtual 10.10.1.1 UDP isakmp
 serverfarm VPN
 sticky 3600 group 69
 inservice
```

Examples How to Configure IOS SLB with RADIUS Load Balancing

This section contains the following examples, illustrating several different IOS SLB RADIUS load-balancing configurations:

Example How to Configure IOS SLB with RADIUS Load Balancing for a GPRS Network

The figure below shows a typical IOS SLB RADIUS load-balancing configuration for a GPRS network. In this configuration:

- RADIUS requests are load-balanced between SSG RADIUS proxy servers 10.10.10.1 and 10.10.10.2.
- End-user data packets are routed to the IOS SLB device.
- End-user data packets from the 1.1.1.0 subnet are directed by IOS SLB to SSG1.

• End-user data packets from the 1.1.2.0 subnet are directed by IOS SLB to SSG2.

Figure 25: IOS SLB with RADIUS Load Balancing for a GPRS Network
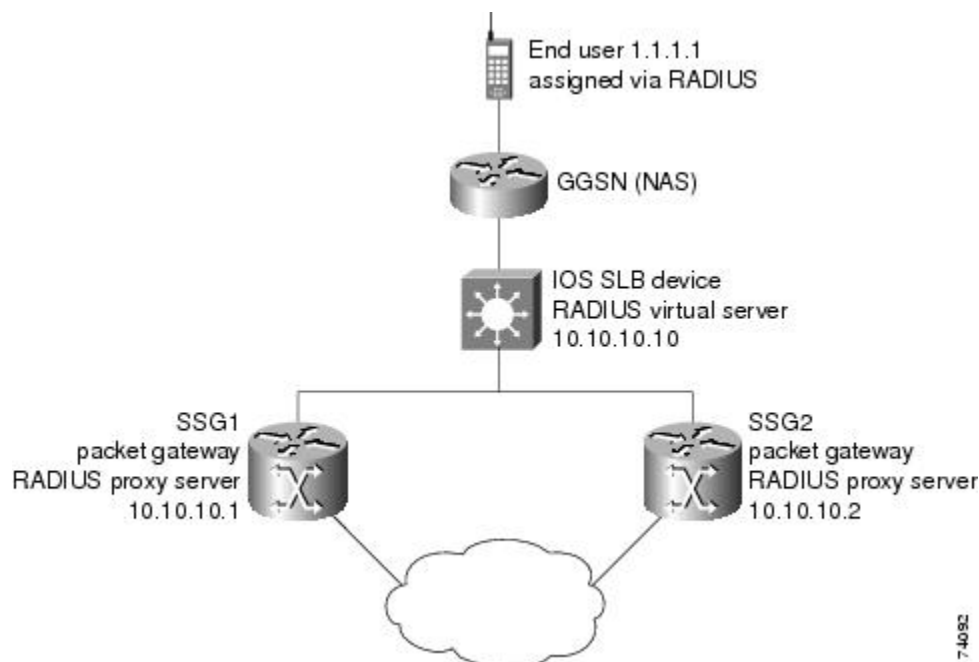
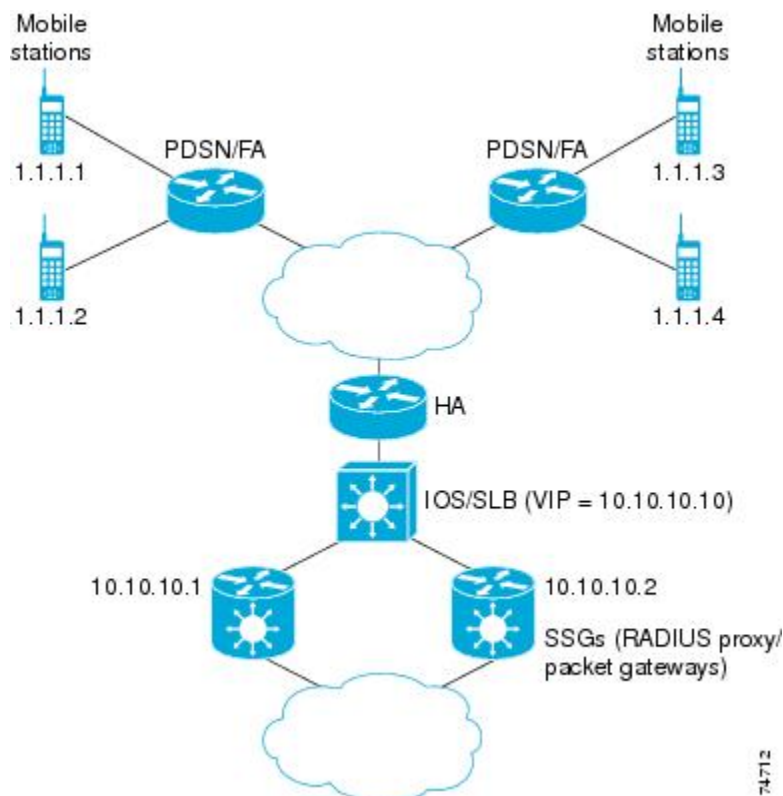Following are the IOS SLB configuration statements for the configuration shown in the figure above:

```
ip slb route 1.1.0.0 255.255.0.0 framed-ip
!
ip slb serverfarm SSGFARM
 nat server
 real 10.10.10.1
  inservice
 real 10.10.10.2
  inservice
!
ip slb vserver RADIUS_ACCT
 virtual 10.10.10.10 udp 1813 service radius
 serverfarm SSGFARM
 idle radius request 20
 idle radius framed-ip 7200
 sticky radius framed-ip group 1
 inservice
!
ip slb vserver RADIUS_AUTH
 virtual 10.10.10.10 udp 1812 service radius
 serverfarm SSGFARM
 idle radius request 20
 idle radius framed-ip 7200
 sticky radius framed-ip group 1
 inservice
```

Example How to Configure IOS SLB with RADIUS Load Balancing for a Simple IP CDMA2000 Network

The figure below shows a typical IOS SLB RADIUS load-balancing configuration for a CDMA2000 network with simple IP service. In this configuration:

- The RADIUS virtual server IP address for the PDSNs is 10.10.10.10.
- RADIUS requests are load-balanced between SSG RADIUS proxy servers 10.10.10.1 and 10.10.10.2.
- End-user data packets are routed to the IOS SLB device.
- End-user data packets from the 1.1.0.0 network are routed to the SSGs.

Figure 26: IOS SLB with RADIUS Load Balancing for a Simple IP CDMA2000 Network

Following are the IOS SLB configuration statements for the configuration shown in the figure above:

```
ip slb route 1.1.0.0 255.255.0.0 framed-ip
!
ip slb serverfarm SSGFARM
  nat server
  real 10.10.10.1
    inservice
  real 10.10.10.2
    inservice
!
ip slb vserver RADIUS_SIP
  virtual 10.10.10.10 udp 0 service radius
  serverfarm SSGFARM
  idle radius framed-ip 3600
  sticky radius username
  sticky radius framed-ip
  inservice
```

Example How to Configure IOS SLB with RADIUS Load Balancing for a Mobile IP CDMA2000 Network

The figure below shows a typical IOS SLB RADIUS load-balancing configuration for a CDMA2000 network with Mobile IP service. In this configuration:

- The RADIUS virtual server IP address for the PDSNs and the HA is 10.10.10.10.
- RADIUS requests are load-balanced between SSG RADIUS proxy servers 10.10.10.1 and 10.10.10.2.
- End-user data packets are routed to the IOS SLB device.
- End-user data packets from the 1.1.0.0 network are routed to the SSGs.

Figure 27: IOS SLB with RADIUS Load Balancing for a Mobile IP CDMA2000 Network

Following are the IOS SLB configuration statements for the configuration shown in the figure above:

```
ip slb route 1.1.0.0 255.255.0.0 framed-ip
!
ip slb serverfarm SSGFARM
  nat server
  real 10.10.10.1
    inservice
  real 10.10.10.2
    inservice
!
ip slb vserver RADIUS_SIP
  virtual 10.10.10.10 udp 0 service radius
  serverfarm SSGFARM
  idle radius framed-ip 3600
  sticky radius username
  sticky radius framed-ip
  inservice
```

Example How to Configure IOS SLB with RADIUS Load Balancing for Multiple Service Gateway Farms

The following sample configuration enables IOS SLB to balance packet flows for a set of subscribers among multiple service gateway server farms (in this sample, a server farm of SSGs and a server farm of CSGs):

```
ip slb route 1.1.0.0 255.255.0.0 framed-ip
!
ip slb serverfarm SSGFARM
 nat server
 real 10.10.10.1
  inservice
 real 10.10.10.2
  inservice
!
ip slb serverfarm CSGFARM
 nat server
 real 20.20.20.1
  inservice
 real 20.20.20.2
  inservice
!
ip slb vserver SSG_AUTH
 virtual 10.10.10.10 udp 1812 service radius
 serverfarm SSGFARM
 idle radius request 20
 idle radius framed-ip 7200
 sticky radius framed-ip group 1
 access Vlan20 route framed-ip
 inservice
!
ip slb vserver SSG_ACCT
 virtual 10.10.10.10 udp 1813 service radius
 serverfarm SSGFARM
 idle radius request 20
 idle radius framed-ip 7200
 sticky radius framed-ip group 1
 access Vlan20 route framed-ip
 inservice
!
ip slb vserver CSG_ACCT
 virtual 20.20.20.20 udp 1813 service radius
 serverfarm CSGFARM
 idle radius request 25
 idle radius framed-ip 0
 sticky radius framed-ip
 access Vlan30 route framed-ip
 inservice
```

Example How to Configure IOS SLB with RADIUS Load Balancing Firewall Load Balancing "Sandwich"

The figure below shows a RADIUS load balancing/firewall load balancing "sandwich" on one IOS SLB device. In this sample configuration:

- The RADIUS load balancing virtual IP address is 5.5.5.5.
- The subscriber framed-IP network is 1.0.0.0/255.0.0.0.
- VL105, VL106, VL107, and VL108 are VLANs.
- RADIUS requests arriving on VLAN VL105 are balanced to 10.10.106.42 and 10.10.106.43.
- User traffic is stickied based on framed-IP address assignments in the 1.0.0.0 subnet.

- Firewall load balancing on the other side (10.10.107.42/43) ensures that return path traffic to the subscriber is delivered to the correct gateway.

Figure 28: IOS SLB with RADIUS Load Balancing/Firewall Load Balancing "Sandwich"
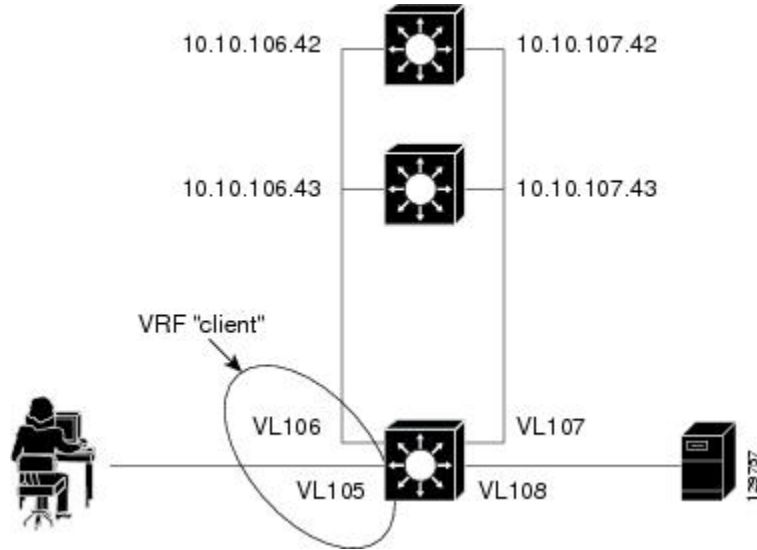
Following are the IOS SLB configuration statements for the configuration shown in the figure above:

```
ip vrf client
 rd 0:1
!
ip slb probe P742 ping
 address 10.10.107.42
 interval 120
!
ip slb probe P743 ping
 address 10.10.107.43
 interval 120
!
ip slb route 1.0.0.0 255.0.0.0 framed-ip
ip slb route framed-ip deny
!
ip slb firewallfarm SERVER
 access inbound Vlan108
 access outbound Vlan107
 inservice
 real 10.10.107.42
  probe P742
  inservice
 real 10.10.107.43
  probe P743
  inservice
 protocol tcp
  sticky 180 destination
 protocol datagram
  sticky 180 destination
 predictor hash address port
!
ip slb serverfarm SF1
 nat server
 access Vlan106
!
 real 10.10.106.42
  inservice
 real 10.10.106.43
  inservice
```

```
!
ip slb vserver VS1
 virtual 5.5.5.5 udp 0 service radius
 serverfarm SF1
 sticky radius framed-ip
 access Vlan105 route framed-ip
 access Vlan105
 inservice
!
mls flow ip interface-full
!
!*************************************************
!* Switchports, port channels and trunks        *
!* added to vlans 105-108 (left out for brevity) *
!*************************************************
!
interface Vlan105
 ip vrf forwarding client
 ip address 10.10.105.2 255.255.255.0
!
interface Vlan106
 ip vrf forwarding client
 ip address 10.10.106.2 255.255.255.0
!
interface Vlan107
 ip address 10.10.107.2 255.255.255.0
!
interface Vlan108
 ip address 10.10.108.2 255.255.255.0
!
ip route 10.10.105.0 255.255.255.0 10.10.107.42
ip route vrf client 10.10.108.0 255.255.255.0 10.10.106.42
```

Example How to Configure IOS SLB with RADIUS Load Balancing Maps

The following sample configuration enables IOS SLB to support multiple server farms behind a RADIUS load balancing virtual server, using RADIUS calling station IDs and usernames to select server farms. Server farm **farm3** is configured without an associated map, and therefore acts as a default server farm. If IOS SLB cannot match any of the other server farm maps, and a default server farm is configured, IOS SLB sends the RADIUS request to the default serverfarm.

```
ip slb serverfarm CSGFARM
 predictor route-map rlb-pbr
ip slb serverfarm AAAFARM
 nat server
 real 10.10.10.1
  inservice
 real 10.10.10.2
  inservice
ip slb vserver RADIUS_ACCT
 virtual 10.10.10.10 udp 1813 service radius
 serverfarm CSGFARM
 radius inject acct 1 key 0 cisco
 inservice
ip slb vserver RADIUS_AUTH
 virtual 10.10.10.10 udp 1812 service radius
 serverfarm AAAFARM
 radius inject auth 1 calling-station-id
 radius inject auth timer 45
 radius inject auth vsa cisco
 inservice
!
interface vlan 100
 ip policy route-map rlb-pbr
!
access-list 1 permit 0.0.0.1 255.255.255.254
access-list 2 permit 0.0.0.0 255.255.255.254
!
route-map rlb-pbr permit 10
```

```
 match ip address 1
 set ip next-hop 10.10.10.1
!
route-map rlb-pbr permit 20
 match ip address 2
 set ip next-hop 10.10.10.2
```
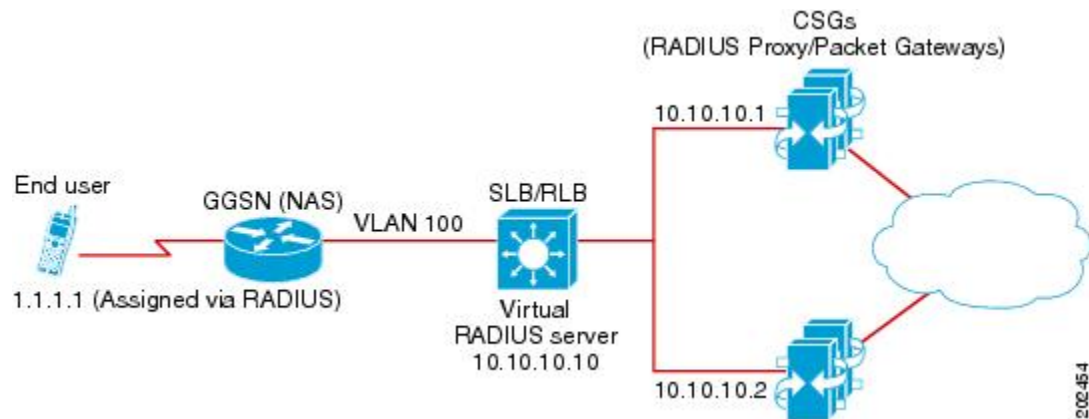
Example How to Configure IOS SLB with RADIUS Load Balancing Accelerated Data Plane Forwarding

The following IOS SLB configuration has the following characteristics:

- There is a virtual RADIUS server with IP address 10.10.10.10 that manages Network Access Server (NAS) devices.
- There are two packet gateways with IP addresses 10.10.10.1 and 10.10.10.2.
- RADIUS traffic destined for the virtual RADIUS server is distributed between the packet gateways, based on mapped framed-IP addresses, according to route map **rlb-pbr**.
- Server farm CSGFARM is configured with real IP addresses that match the possible results for route map **rlb-pbr**.
- End user traffic arriving on VLAN 100 is routed to the correct Cisco Content Services Gateway (CSG) based on access control lists (ACLs):

    ◦ ACL 1 sends IP addresses ending in odd numbers to the CSGs behind packet gateway 10.10.10.1.
    ◦ ACL 2 sends IP addresses ending in even numbers to the CSGs behind packet gateway 10.10.10.2.

Figure 29: IOS SLB with RADIUS Load Balancing Accelerated Data Plane Forwarding

Following are the IOS SLB configuration statements for the configuration shown in the figure above:

```
ip slb serverfarm CSGFARM
 predictor route-map rlb-pbr
ip slb serverfarm AAAFARM
 nat server
 real 10.10.10.1
  inservice
 real 10.10.10.2
  inservice
!
ip slb vserver RADIUS_ACCT
 virtual 10.10.10.10 udp 1813 service radius
 serverfarm CSGFARM
 radius inject acct 1 key 0 cisco
```

```
 inservice
!
ip slb vserver RADIUS_AUTH
 virtual 10.10.10.10 udp 1812 service radius
 serverfarm AAAFARM
 radius inject auth 1 calling-station-id
 radius inject auth timer 45
 radius inject auth vsa cisco
 inservice
!
interface vlan 100
 ip policy route-map rlb-pbr
!
access-list 1 permit 0.0.0.1 255.255.255.254
access-list 2 permit 0.0.0.0 255.255.255.254
!
route-map rlb-pbr permit 10
 match ip address 1
 set ip next-hop 10.10.10.1
!
route-map rlb-pbr permit 20
 match ip address 2
 set ip next-hop 10.10.10.2
```

Example How to Configure IOS SLB with Home Agent Director

The following sample configuration enables IOS SLB to balance Mobile IP RRQs among multiple home agents.

Figure 30: IOS SLB with Home Agent Director
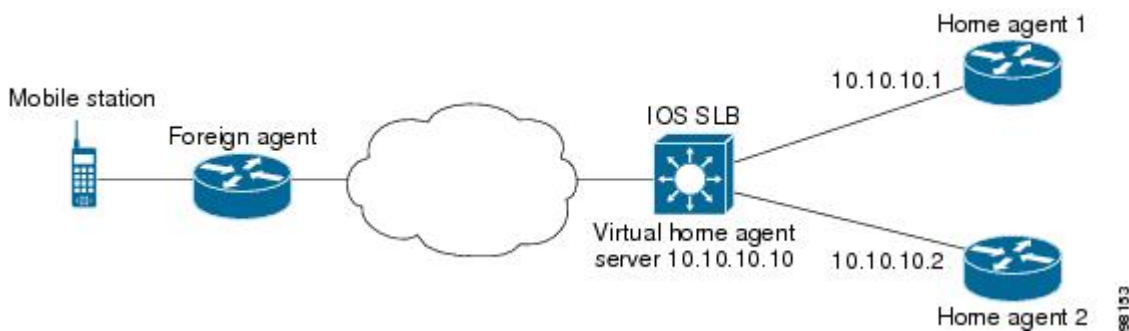
Following are the IOS SLB configuration statements for the configuration shown in the figure above:

```
ip slb serverfarm HA_FARM
 nat server
 real 10.10.10.1
  inservice
 real 10.10.10.2
  inservice
ip slb vserver VIRTUAL_HA
 virtual 10.10.10.10 udp 434 service ipmobile
 serverfarm HA_FARM
 inservice
```

Example How to Configure IOS SLB with Sticky Connections

The following sample configuration assigns all HTTP connections from a subnet to the same real server in server farm PUBLIC:

```
ip slb vserver http
  serverfarm PUBLIC
  sticky 30 group 1 netmask 255.255.255.248
  virtual 20.20.20.20 tcp 80
  inservice
```

The following sample configuration adds HTTPS connections to the above configuration, using the same sticky information but with a different virtual server:

```
ip slb vserver https
  serverfarm PUBLIC
  sticky 30 group 1 netmask 255.255.255.248
  virtual 20.20.20.20 tcp 443
  inservice
```

Now, all HTTP *and* HTTPS connections from the subnet are assigned to the same real server. For example, if a user connects to HTTP, then a second user connects to HTTPS, both connections are assigned to the same real server.

Example How to Configure IOS SLB with GTP IMSI Sticky Database

The following sample configuration shows how to enable the IOS SLB GTP IMSI sticky database:

```
ip slb serverfarm GGSN_FARM
 failaction gtp purge
 real 10.20.10.1
  weight 1
  faildetect numconns 255 numclients 8
  inservice
!
 real 10.20.10.2
  weight 1
  faildetect numconns 255 numclients 8
  inservice
!
 real 10.20.10.3
  weight 1
  faildetect numconns 255 numclients 8
  inservice
!
ip slb vserver GGSN_SERVER1
 virtual 10.10.10.10 udp 3386 service gtp
 serverfarm GGSN_FARM backup GGSN_FARM
 idle gtp request 90
 idle gtp imsi 10000000
 sticky gtp imsi group 1
 gtp notification cac 3
 inservice
!
ip slb vserver GGSN_SERVER2
 virtual 10.10.10.10 udp 2123 service gtp
 serverfarm GGSN_FARM backup GGSN_FARM
 idle gtp request 90
 idle gtp imsi 10000000
 sticky gtp imsi group 1
```

```
gtp notification cac 3
inservice
```

Example How to Configure IOS SLB with ASN Sticky Database

The following sample configuration shows how to enable the IOS SLB ASN sticky database:

```
ip slb entries sticky 15000 800000
ip slb serverfarm ASNLB_FARM
 failaction asn purge
!
 real 10.20.10.1
  weight 1
  faildetect numconns 255 numclients 8
  inservice
!
 real 10.20.10.2
  weight 1
  faildetect numconns 255 numclients 8
  inservice
!
 real 10.20.10.3
  weight 1
  faildetect numconns 255 numclients 8
  inservice
!
ip slb vserver ASNLB_SERVER
 virtual 10.10.10.10 udp 0 service asn
 serverfarm ASNLB_FARM
 idle asn request 90
 idle asn msid 100000
 sticky asn msid group 1
 gw port 63082
 replicate casa 100.100.100.102 100.100.100.101 1024 password hello
 inservice
```

Example How to Configure IOS SLB with Transparent Web Cache Load Balancing

In the following sample configuration, virtual server WEBCACHE examines all web flows passing through the load-balancing device and dispatches them to server farm WEBCACHE-FARM. The **client exclude** statement ignores flows originating from subnet 80.80.7.0, enabling the real servers 80.80.7.188 and 80.80.7.189 to communicate with the Internet as needed.

```
ip slb serverfarm WEBCACHE-FARM
  real 80.80.7.188
   inservice
  real 80.80.7.189
   inservice
ip slb vserver WEBCACHE
  virtual 0.0.0.0 0.0.0.0 tcp www
  serverfarm WEBCACHE-FARM
  client 80.80.7.0 255.255.255.0 exclude
  inservice
```

Example How to Configure IOS SLB with KAL-AP Agent

In the following sample configuration, the client is configured to send a Domain Name System (DNS) query **abcd.com** to the GSS. The Global Site Selector (GSS) in the DUBLIN site receives the requests from the client. The GSS answers the DNS query with the virtual IP address of either CHICAGO (10.0.0.100) or NEWYORK (10.0.0.200), based on the load reported by those virtual servers.

Figure 31: IOS SLB with KAL-AP Agent
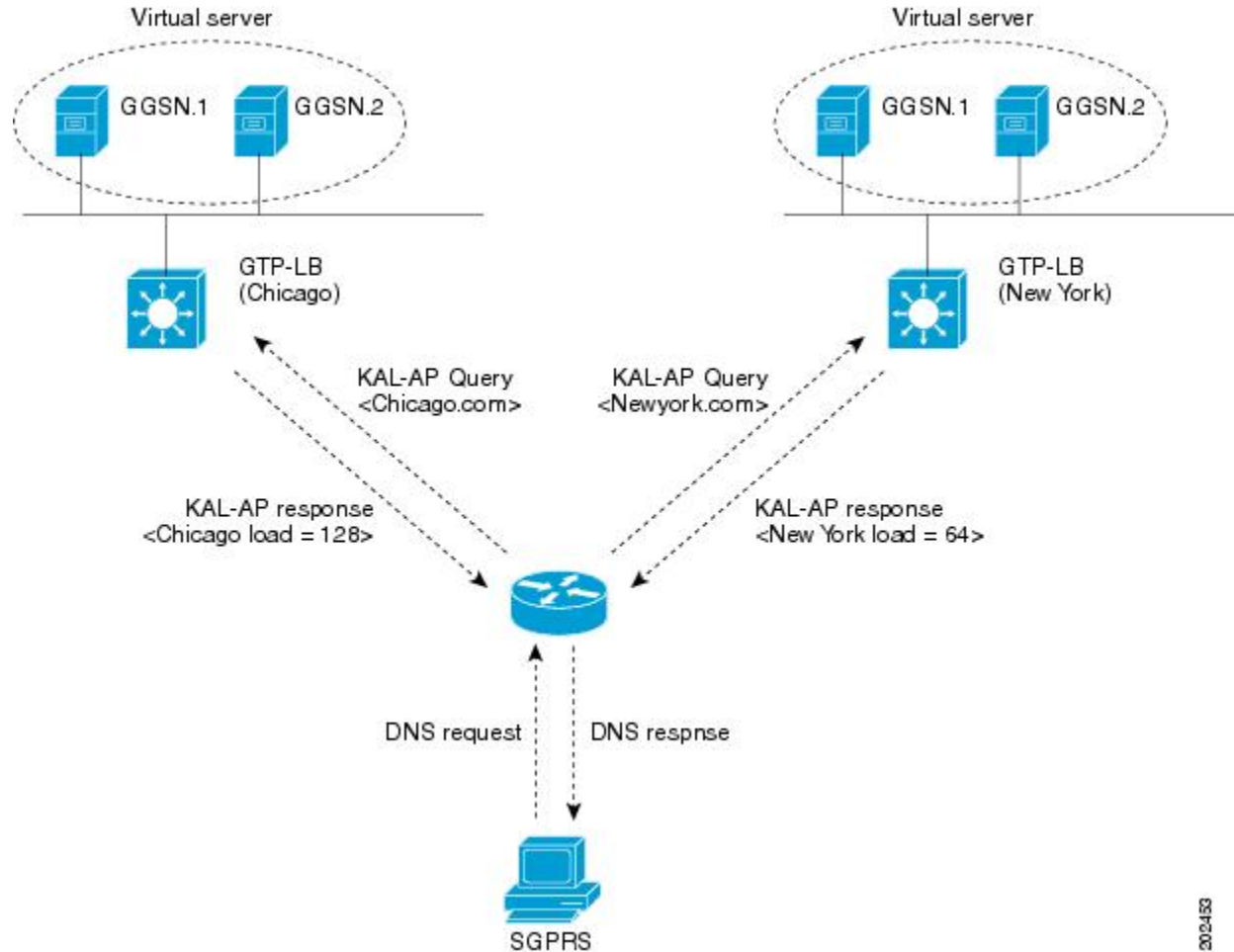
Following are the IOS SLB configuration statements for the configuration shown in the figure above:

GSS

```
shared-keepalive kalap 192.168.1.1 capp-secure enable key kap
shared-keepalive kalap 192.168.2.1 capp-secure enable key kap
```

```
!
answer vip 10.0.0.100 name CHICAGO activate
 keepalive type kalap tag 192.168.1.1 chicao.com
answer vip 10.0.0.200 name NEWYORK activate
 keepalive type kalap tag 192.168.2.1 newyork.com
!
answer-group ABCD owner System type vip
answer-add 10.0.0.100 name CHICAGO weight 1 order 0 load-threshold 254 activate
answer-add 10.0.0.200 name NEWYORK weight 1 order 0 load-threshold 254 activate
dns rule ABCDGPRS owner System source-address-list Anywhere domain-list abcd.com query a
 clause 1 vip-group    method least-loaded ttl 20 count 1 sticky disable
```

Site-1 IOS SLB - CHICAGO

```
ip slb capp udp
 peer port 6000 secret 0 kap
!
ip slb serverfarm SF
 kal-ap domain chicago.com
 farm-weight 200
 real 10.10.10.1
  inservice
 real 10.10.10.2
  inservice
!
ip slb vserver chicago
 virtual 10.0.0.100 udp 0
 serverfarm SF
 inservice
!
ip slb dfp
 agent 10.10.10.1 5000 30 0 10
 agent 10.10.10.2 5000 30 0 10
!
int vlan100
 ip address 192.168.1.1 255.255.255.0
```

GGSN-1

```
gprs dfp max-weight 100
gprs maximum-pdp-context-allowed 20000
!
ip dfp agent gprs
 port 5000
 inservice
```

GGSN-2

```
gprs dfp max-weight 100
gprs maximum-pdp-context-allowed 20000
!
ip dfp agent gprs
 port 5000
 inservice
```

Site-2 IOS SLB - NEWYORK

```
ip slb capp udp
 peer port 6000
```

```
 peer 192.1.1.1 secret 0 test
 peer 10.100.100.100 port 1234
!
ip slb serverfarm SF
 kal-ap domain newyork.com
 farm-weight 6200
 real 10.20.20.1
  inservice
 real 10.20.20.2
  inservice
 real 10.20.20.3
  inservice
 real 10.20.20.4
  inservice
!
ip slb vserver chicago
 virtual 10.0.0.200 udp 0
 serverfarm SF
 inservice
!
ip slb dfp
 agent 10.10.10.1 5000 30 0 10
 agent 10.10.10.2 5000 30 0 10
!
int vlan200
 ip address 192.168.2.1 255.255.255.0
```

GGSN-1

```
gprs dfp max-weight 100
gprs maximum-pdp-context-allowed 20000
!
ip dfp agent gprs
 port 5000
 inservice
```

GGSN-2

```
gprs dfp max-weight 100
gprs maximum-pdp-context-allowed 20000
!
ip dfp agent gprs
 port 5000
 inservice
```

Troubleshooting IOS SLB

Question	Answer
Can I use IOS SLB to load-balance clients and real servers that are on the same LAN or VLAN?	NO! IOS SLB does not support load balancing of flows between clients and real servers that are on the same LAN or VLAN. The packets being load-balanced cannot enter and leave the load-balancing device on the same interface.
Why is IOS SLB not marking my connections as ESTABLISHED even though I'm transferring data?	If you are using dispatched mode, make sure there are no alternate paths that allow outbound flows to bypass IOS SLB. Also, make sure the clients and real servers are not on the same IP subnet (that is, they are not on the same LAN or VLAN).
Why am I able to connect to real servers directly, but unable to connect to the virtual server?	Make sure that the virtual IP address is configured as a loopback in each of the real servers (if you are running in dispatched mode).
Why is IOS SLB not marking my real server as failed when I disconnect it from the network?	Tune the values for the **numclients**, **numconns**, and **delay** keywords. If you have a very small client population (for example, in a test environment), the **numclients** keyword could be causing the problem. This parameter prevents IOS SLB from mistaking the failure of a small number of clients for the failure of a real server.
Why does IOS SLB show my real server as INSERVICE even though I have taken it down or physically disconnected it?	The INSERVICE and OUTOFSERVICE states indicate whether the network administrator *intends* for that real server to be used when it is operational. A real server that was INSERVICE but was removed from the selection list dynamically by IOS SLB as a result of automatic failure detection, is marked as FAILED. Use the **show ip slb reals detail** command to display these real server states.

Question	Answer
	Beginning with release 12.1(1)E, INSERVICE is changed to OPERATIONAL, to better reflect what is actually occurring.
How can I verify that IOS SLB sticky connections are working properly?	Use the following procedure: 1 Configure the sticky connections. 2 Start a client connection. 3 Enter the **show ip slb reals detail** and **show ip slb conns** commands. 4 Examine the real server connection counts. The real server whose count increased is the one to which the client connection is assigned. 5 Enter the **show ip slb sticky** command to display the sticky relationships stored by IOS SLB. 6 End the connection. 7 Ensure that the real server's connection count decreased. 8 Restart the connection, after waiting no longer than the sticky timeout value. 9 Enter the **show ip slb conns** command again. 10 Examine the real server connection counts again, and verify that the sticky connection is assigned to the same real server as before.

How can I verify that server failures are being detected correctly?	Use the following procedure:
	
	1 Use a large client population. If the number of clients is very small, tune the **numclients** keyword on the **faildetect numconns (real server)** command so that the servers are not displayed as FAILED.
	2 Enter the **show ip slb reals detail** command to show the status of the real servers.
	3 Examine the status and connection counts of the real servers.
	
	    • Servers that failed show a status of FAILED, TESTING, or READY_TO_TEST, based on whether IOS SLB is checking that the server came back up when the command was sent.
	    • When a real server fails, connections that are assigned but not established (no SYN or ACK is received) are reassigned to another real server on the first inbound SYN after the **reassign** threshold is met.

Question	Answer
	However, any connections that were already established are forwarded to the same real server because, while it might not be accepting new connections, it might be servicing existing ones.
	• For weighted least connections, a real server that has just been placed in service starts slowly so that it is not overloaded with new connections. (See the "Slow Start" section for more information.) Therefore, the connection counts displayed for a new real server show connections going to other real servers (despite the new real server's lower count). The connection counts also show "dummy connections" to the new real server, which IOS SLB uses to artificially inflate the connection counts for the real server during the slow start period.
Does the **no inservice** command take a resource out of service immediately?	When you use the **no** form of the **inservice** command to remove a firewall, firewall farm, real server, or virtual server from service, the resource acquiesces gracefully. No new connections are assigned, and existing connections are allowed to complete. To stop all existing connections for an entire firewall farm or virtual server immediately, use the **clear ip slb connections** command.
I configured both IOS SLB and input ACLs on the same Catalyst 6500 Family Switch, and now I see TCAM Capacity Exceeded messages. Why?	If you configure IOS SLB and either input ACLs or firewall load balancing on the same Catalyst 6500 Family Switch, you can exceed the capacity of the Telecommunications Access Method (TCAM) on the Policy Feature Card (PFC). To correct the problem, use the mls ip slb search wildcard rp command to reduce the amount of TCAM space used by IOS SLB, but be aware that this command can result in a slight increase in route processor use.
Which IOS releases and platforms support IOS SLB VRF?	Virtual Private Network (VPN) routing and forwarding (VRF) for IOS SLB is supported in IOS release 12.2(18)SXE or later on the Supervisor Engine 720 with an MSFC3 (SUP720-MSFC3) for the Cisco 7600 series routers.

Question	Answer
What can cause IOS SLB out-of-sync messages on the Supervisor?	If you are using one Supervisor Engine with **replicate slave** configured, you might receive out-of-sync messages on the Supervisor.
Can IOS SLB provide both firewall load balancing and RADIUS load balancing on the same Supervisor?	IOS SLB can provide both firewall load balancing and RADIUS load balancing on the same Supervisor Engine 720 (SUP720-MSFC3).

Additional References for IOS SLB

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS configuration fundamentals	*Cisco IOS Configuration Fundamentals Configuration Guide*
DFP configuration information	*Dynamic Feedback Protocol Support in Distributed Director*
CFM configuration information	*Using Content Flow Monitor*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
CISCO-SLB-MIB CISCO-SLB-CAPABILITY **Note** Although the objects in these MIBs are defined as *read-create* , you cannot use the SNMP SET command to modify them. Instead, you must use the command line to set the associated command line keywords, after which the new values are reflected in SNMP.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1631	The IP Network Address Translator (NAT)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html