



# Server Load Balancing

---

This chapter describes the Server Load Balancing (SLB) feature and contains the following sections:

- SLB Overview, page 5-1
- Required SLB Configuration Tasks, page 5-13
- Optional SLB Configuration Tasks, page 5-19
- Required URL-Based SLB Configuration Tasks, page 5-29
- Optional URL-Based SLB Configuration Tasks, page 5-33
- Monitoring and Maintaining SLB, page 5-34



Note

---

You are at Step 3 in the suggested process for configuring your switch. See the “Switch Configuration Steps” section on page 2-1. You should have already configured the Catalyst 4840G processor and now be ready to proceed with configuring SLB.

---

## SLB Overview

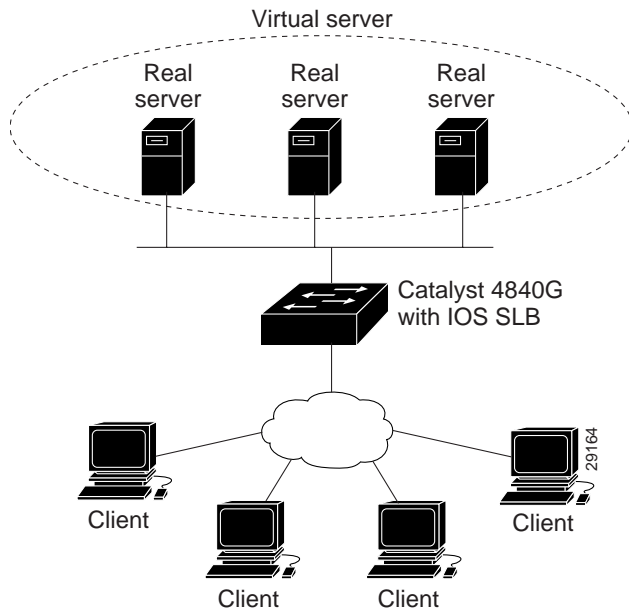
The SLB feature is an IOS-based solution that provides IP server load balancing. Using this feature, you can define a *virtual* server which represents a group of *real* servers in a cluster of network servers known as a *server farm*. In this environment a client connects to the IP address of the virtual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

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

SLB provides server takeover capabilities in failure cases where one of the real servers goes out of service due to a scheduled or unscheduled outage. In the event of a real server failure, an available alternate server takes over the functions of the failed server with minimal disruption to the client.

Figure 5-1 illustrates an SLB network.

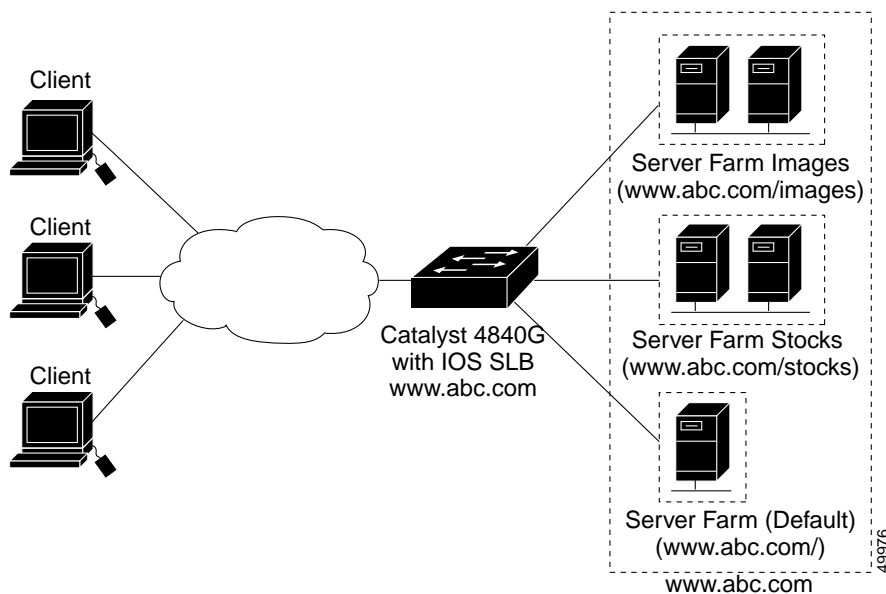
**Figure 5-1 Logical View of SLB**



SLB supports URL-based load balancing. URL-based SLB is an IOS-based solution that provides control over load balancing based on a URL. URL-based load balancing provides a way to intercept URL requests directed to a particular Web server and to distribute the requests across a set of server farms. Any server belonging to the server farm is capable of processing the request sent to the particular URL. The matching criteria used are exact match, longest match, prefix match, and suffix match.

Figure 5-2 illustrates a URL-based SLB network.

**Figure 5-2 Logical View of URL-Based SLB**



## Configurable SLB Features

This section describes the features you can configure in SLB:

- Algorithms for SLB, page 5-3
- Alternate IP Addresses, page 5-4
- Automatic Server Failure Detection, page 5-4
- Automatic Unfail, page 5-4
- Client-Assigned Load Balancing, page 5-4
- Content Flow Monitor Support, page 5-5
- Delayed Removal of TCP Connection Context, page 5-5
- Dynamic Feedback Protocol for SLB, page 5-5
- HTTP Redirect Load Balancing, page 5-5
- Maximum Connections, page 5-6
- Network Address Translation (NAT), page 5-6
- Port-Bound Servers, page 5-8
- Probes, page 5-8
- Redundancy Enhancements, page 5-8
- Slow Start, page 5-9
- Sticky Connections, page 5-9
- SynGuard, page 5-9
- TCP Session Reassignment, page 5-9
- Transparent Web Caches Balancing, page 5-10

### Algorithms for SLB

SLB provides two load-balancing algorithms: weighted round robin and weighted least connections. You can specify either algorithm as the basis for choosing a real server for each new connection request that arrives at the virtual server.

#### Weighted Round Robin

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

For example, assume a server farm consisting of real server Server A with  $n = 3$ , Server B with  $n = 1$ , and Server C with  $n = 2$ . The first three connections to the virtual server are assigned to Server A, the fourth connection to Server B, and the fifth and sixth connections to Server C.

**Note**

---

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

---

## Weighted Least Connections

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

For example, assume a server farm consisting of real server Server A with  $n = 3$ , Server B with  $n = 1$ , and Server C with  $n = 2$ . Server A would have a calculated capacity of  $3/(3+1+2)$ , or half of all active connections on the virtual server, Server B one-sixth of all active connections, and Server C 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 switch to use an unweighted least connections algorithm.

---

## Alternate IP Addresses

SLB allows you to use Telnet to connect to the load-balancing device by using an alternate IP address. You can use either of the following methods:

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

This function is similar to the function provided by the LocalDirector (LD) **alias** command.

## Automatic Server Failure Detection

SLB automatically detects each failed attempt to connect to a real server by means of TCP. SLB then 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.

## 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 time expires, the server is again eligible for new virtual server connections, and 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.

## Client-Assigned Load Balancing

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

## Content Flow Monitor Support

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, SLB might “see” the termination of a TCP connection (a finish [FIN] or reset [RST]) followed by other packets for the connection. This problem usually occurs when there are multiple paths that the TCP connection packets can follow. To correctly redirect the packets that arrive after the connection is terminated, 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 is terminated is controlled by a configurable *delay timer*.

## Dynamic Feedback Protocol for SLB

SLB Dynamic Feedback Protocol (DFP) is a mechanism that allows host agents in load-balanced environments to dynamically report the change in status of the host systems that provide a virtual service. The status reported is a relative weight that specifies a host server’s capacity to perform work.

When a DFP agent is defined on SLB, a TCP connection is initiated from the manager to the DFP agent. Once this connection is established, the agent periodically sends update information across the connection to SLB. This information is used by SLB as an aid in load balancing the real servers, as well as acting as an application-level keepalive for the connection. If an agent on the real server has no information to send, and an inactivity timeout was specified for this DFP agent, the agent must send an empty report to prevent termination of the connection. In the event of a failure, SLB uses a default weight for the real servers.

The DFP agent can be either software running on the real server or a separate unit that collects information from one or more real servers. The DFP agent consolidates the information, formats it into DFP, and reports the information to SLB at periodic intervals. If a need arises, such as a sudden change in a real server’s ability to handle traffic, the DFP agent can send an early report.

Cisco Systems DistributedDirector can use DFP to obtain availability information on virtual servers from SLB. DistributedDirectors can use this information to identify load imbalances over multiple sites and distribute Internet traffic more evenly. The process is performed by a DFP manager on the DistributedDirector and a DFP agent on the SLB device. The DFP agent calculates an availability metric for the virtual servers, and the DFP manager uses the metric to make load distribution decisions.

## HTTP Redirect Load Balancing

The HTTP redirect load balancing feature provides persistence across HTTP and SSL sessions. All HTTP requests that belong to the same transaction are directed to the same real server. An example of such a transaction is a user’s shopping on the Web by adding items to a “shopping cart” and then electronically purchases them.

HTTP permits a Web server to redirect HTTP requests to another server. Using HTTP redirect load balancing, the Web server is identified by a public virtual IP address known as the Web Server Virtual Address (WSVA). When HTTP redirect load balancing is configured, SLB intercepts HTTP requests to the Web server and returns the IP address of an intermediate virtual server (a Redirected Server Virtual Address [RSVA]) to the originator of the HTTP request. The RSVA corresponds to a real server. After

being redirected, the client then sends connection requests to the IP address of the RSVA instead of the WSVA. Because the SLB device maps an RSVA IP address to a single real server, persistence is created across the connections between the client and the real server.

The HTTP redirect load-balancing feature provides high performance by minimizing the number of TCP connections that must be proxied. This feature also monitors TCP connection events and therefore can quickly detect a real server failure. In case of server failure, packets are redirected to another RSVA or to a specified backup location if it is a new connection, or the connection is reset if it already exists.

## Maximum Connections

SLB allows you to configure *maximum connections* for each server.

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 that server, SLB automatically switches all further connection requests to another server until the connection number drops below the specified limit.

## Network Address Translation (NAT)

Cisco IOS NAT, RFC 1631, allows unregistered private IP addresses to connect to the Internet by translating them into globally registered IP addresses. NAT also increases network privacy by hiding internal IP addresses from external networks.

SLB can operate in one of two redirection modes:

- **Dispatched mode**—The virtual server address is known to the real servers, and SLB redirects packets to the real servers at the media access control (MAC) layer. Since the virtual server IP address is not modified in dispatched mode, the real servers must be Layer 2-adjacent to the SLB device; if not, then intervening routers might not be able to route data to the chosen real server.




---

**Note** SLB supports FTP load balancing only in dispatched mode. Therefore, FTP cannot use NAT.

---

- **Directed mode**—The virtual server can be assigned an IP address that is not known to any of the real servers. SLB translates packets exchanged between a client and real server, translating the virtual server IP address to a real server address through NAT.

SLB supports the following types of NAT:

- **Server NAT**—By replacing the virtual server IP address with the real server IP address (and vice versa), servers can be many hops away from the load-balancing device, and intervening routers can route to them without requiring tunneling. In addition, loopback and secondary interfaces are no longer required on the real server as they were with dispatched mode. A less common form of server NAT is *server port translation*, which involves replacement of a virtual server port. Server port translation does not require server IP address translation, but the two translations can be used together.




---

**Note** If an IP address is configured as a real IP address for a NAT virtual server, you cannot balance connection requests from that address to a different virtual server (whether NAT or dispatched) on the same load-balancing device.

---

The network designer must ensure that outbound packets travel through IOS SLB, using one of the following methods:

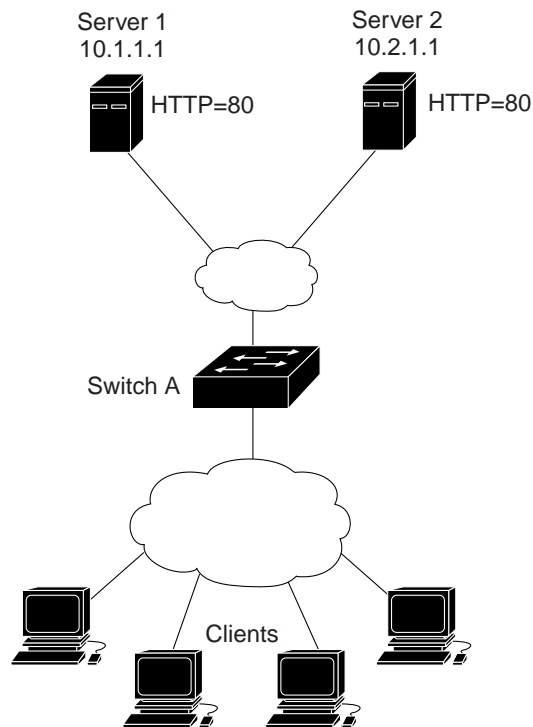
- Direct wiring
- Default gateways or policy-based routing
- SLB NAT of client addresses, enabled as an outbound feature on server-side interfaces
- Client NAT—If multiple load-balancing devices are used, replacing the client IP address with an IP address associated with one of the devices results in proper routing of outbound traffic to the correct device. Client NAT also requires that the ephemeral client port be modified, because many clients can use the same ephemeral port. This allows server NAT to be performed on the packet, and important protocol events (such as TCP SYN, FIN, or RST) are seen by the connection finite state machine. 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.

**Note**

Both server NAT and client NAT are supported for the same connection.

Figure 5-3 shows a sample SLB NAT topology.

**Figure 5-3 Sample NAT Topology**



The topology in Figure 5-3 has two Web servers running a single HTTP server application listening on port 80. Server 1 and Server 2 are load balanced by Switch A, which is performing server address translation.

See the “Configuring SLB Network Address Translation (Optional)” section on page 5-24 to configure SLB NAT on the Catalyst 4840G SLB switch.

## Port-Bound Servers

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

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

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

## Probes

SLB supports both HTTP probes and ping probes. Probes are a simple way to verify connectivity to devices being server load balanced, including devices on the other side of a firewall.

HTTP probes also allow you to monitor applications being server load balanced. With frequent probes the *operation* of each application is verified, not just *connectivity* to the application.

You can configure more than one probe for each server farm.

### 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.

### HTTP Probes

Configure the HTTP probe to expect status code 401; this will eliminate password problems. See the **expect** command for details.

Use the **ip http server** command to configure an HTTP server on the switch. See the description of the **ip http server** command in the *Cisco IOS Configuration Fundamentals Command Reference* for details.

## Redundancy Enhancements

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

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

To reduce that risk, SLB supports two redundancy options, both based on Hot Standby Router Protocol (HSRP):

- Stateless backup—Provides high network availability by routing IP flows from hosts on Ethernet networks, without relying on the availability of a single SLB switch.
- Stateful backup—Enables SLB to incrementally back up its load-balancing decisions (or “keep state”) between primary and backup switches.

SLB also supports *active standby*, by which two SLBs can load balance the same virtual IP address while simultaneously 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 client population to each SLB.
- If a site has more than one virtual IP address, the access router is not needed.

## Slow Start

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

## Sticky Connections

When you use the sticky connections feature, new connections from a client IP address or subnet are assigned to the same real server (for server load balancing), just as previous connections from that address or subnet were assigned.

SLB creates “sticky objects” to track client assignments. These objects remain in the SLB device database after the last sticky connection is deleted, for a period defined by a configurable *sticky timer*. If the timer is configured on a virtual server, new connections from a client are sent to the same real server that handled the previous client connection, provided one of the following is true:

- A connection for the same client already exists.
- The amount of time between the end of a previous connection from the client and the start of a new connection is within the timer duration.

Sticky connections also permit the coupling of services that are handled by more than one virtual server. This allows connection requests for related services to use the same real server. For example, a Web server typically uses TCP port 80 for HTTP, and port 443 for HTTP Secure Socket Layer. 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.

## SynGuard

To prevent a type of network problem known as a *SYN flood denial-of-service attack*, the SynGuard feature limits the rate of TCP SYNs handled by a virtual server. A user might send a large number of SYNs to a server, which could overwhelm the server or cause it to fail, denying service to other users. SynGuard prevents such an attack from bringing down SLB or a real server. SynGuard monitors the number of SYNs to a virtual server over a specific time interval and does not allow the number to exceed a configured *SYN threshold*. If the threshold is reached, any new SYNs are dropped.

## TCP Session Reassignment

SLB tracks each 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*.

## Transparent Web Caches Balancing

You can use SLB to balance transparent Web caches if you know the IP addresses they are serving. To do this, configure the IP addresses (or some common subset of them) as virtual servers.



### Note

A Web cache can start its own connections to real Web sites if pages are not available in its cache. Those connections cannot be load balanced back to the same set of Web caches. SLB addresses this by allowing you to configure “client exclude” statements, so that SLB does not load balance connections initiated by the Web caches.

## SLB Restrictions

SLB has the following restrictions:

- Operates in a standalone mode and currently does not operate as a MultiNode Load Balancing (MNLB) Services Manager. The presence of SLB does not preclude the use of the existing MNLB Forwarding Agent with an external Services Manager in an MNLB environment.
- Does not support coordinating server load-balancing statistics among different SLB instances for backup capability
- Supports FTP only in dispatched mode
- Does not support load-balancing of traffic between clients and real servers that are on the same local area network (LAN) or virtual LAN (VLAN)
- Does not support active standby

## Supported Standards, MIBs, and RFCs

The following standards, MIBs, and RFCs are supported:

- The SLB feature does not support any new or modified standards.
- The CISCO-SLB-MIB supports the SLB feature.
- RFC 1631—Network Address Translator

## Configurable URL-Based SLB Features

This section describes the features you can configure in URL-based SLB:

- URL Map, page 5-11
- Server Farms, page 5-11
- Policies, page 5-11
- Policy-Based Virtual Server, page 5-11
- TCP Parameters, page 5-11
- HTTP 1.1, page 5-12

**Note**

---

URL-based SLB extends the existing SLB function by using all the normal SLB features as well as URL maps, policies, and policy-based virtual servers.

---

## URL Map

URL maps are used to define URLs that are then associated with a URL-based SLB policy. A maximum of four URLs can be configured to a map. The URL configured as part of a URL map is a string containing a sequence of directories, which may terminate with a filename. Within this string the character “/” is the delimiter for each level of a directory. You can configure URLs for longest match, exact match, prefix match, and suffix match.

## Server Farms

A server farm or pool is a collection of servers that have the same content. You specify the server farm name when you add servers to a server farm and when you bind a server pool to a virtual server.

## Policies

A policy links a URL map to a server farm. The order in which policies are linked to a virtual server determines the precedence of the policy. There is no limit to the number of policies that can be created on a Catalyst 4840G SLB switch. When two or more policies match a requested URL, the policy with the highest precedence will be selected.

**Note**

---

All virtual servers configured for URL-based SLB must have a default server farm.

---

## Policy-Based Virtual Server

You can designate a policy-based virtual server to represent a group of server farms with policies associated with each server farm. In this environment the client connects to the IP address of the policy-based virtual server. When the client initiates a connection to the policy-based virtual server, the URL-based SLB function parses the URL request and selects the appropriate server farm, based on user-configured policies. The existing SLB function then chooses a real server for the connection, based on a configured load-balancing algorithm.

## TCP Parameters

TCP is a connection-oriented protocol using known protocol messages for activation and deactivation of TCP sessions. In URL-based SLB, the Finite State Machine is used to correlate TCP signals such as SYN, SYN/ACK, FIN, and RST when adding or removing a connection from the connection database. These signals are also used for determining the number of connections per server and for detecting server failure and recovery.

## HTTP 1.1

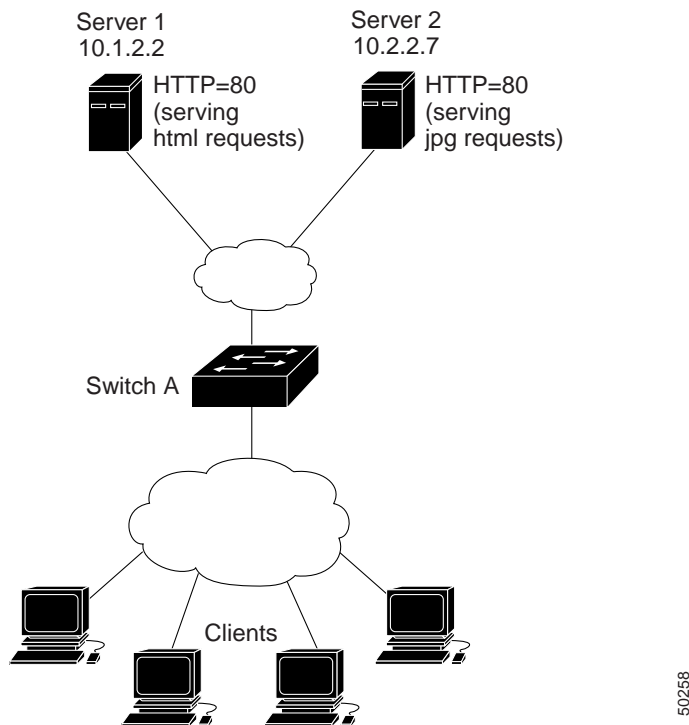
HTTP 1.1 provides you with the ability to have better cache management, chunk encoding, and have persistent connections. When you use HTTP 1.1, you can issue subsequent GET requests without having to open a new connection to the server. You can also pipeline HTTP 1.1's GET requests and request the server to service the GETs in an order different from the one entered.

For HTTP 1.1 to be enabled, URL-based SLB must be configured. The minimum configuration to enable HTTP 1.1 on the Catalyst 4840G SLB switch requires you to:

- Configure two server farms
- Configure a minimum of two URL maps
- Bind the URL maps to the server farms using policies
- Apply the policies to a virtual server farm

Figure 5-4 shows a sample URL-based SLB HTTP 1.1 topology.

**Figure 5-4 Sample HTTP 1.1 Topology**



The topology in Figure 5-4 shows two server farms running a single HTTP server application listening on port 80. Server 1 and Server 2 are load balanced using Switch A, which is performing server address translation. This topology would allow the following scenario to be accomplished in a way that would be invisible to the client issuing the GET requests:

- Configure two server farms. In Figure 5-4 the server farms are designated as Server 1 and Server 2.
- Configure a minimum of two URL maps. Let these maps be M1 and M2. Assume that map M1 matches all the html files (\*.html) and map M2 matches all the jpg objects (\*.jpg).

- Bind the URL maps to the server farms using policies. Let policy P1 bind map M1 to Server 1, and policy P2 bind map M2 to Server 2.
- Apply the policies to a virtual server farm.

In this scenario a GET request for an HTML file would be directed to Server 1. A subsequent GET request for a JPEG object would be redirected to Server 2.

## URL-Based SLB Restrictions

URL-based SLB has the following restrictions:

- The asterisk (\*) wildcard is supported when used in conjunction with prefix and suffix matching.
- The HTTP method of GET is supported.
- HTTP 1.1 pipelining is not supported.

## Required SLB Configuration Tasks

This section describes the tasks necessary to configure a basic SLB network on the Catalyst 4840G SLB switch.

Configuring SLB involves identifying server farms, configuring groups of real servers in server farms, and configuring the virtual servers by which the real servers are represented to the clients. See the following sections for required configuration tasks for the SLB feature:

- Sample SLB Device Network, page 5-13
- Configuring Server Farms, page 5-15
- Verifying Server Farms, page 5-15
- Configuring Virtual Servers, page 5-16
- Verifying Virtual Servers, page 5-16
- Configuring Restricted Clients, page 5-17
- Verifying Restricted Clients, page 5-17
- Verifying SLB Connectivity, page 5-18
- Sample SLB Network Configuration, page 5-18

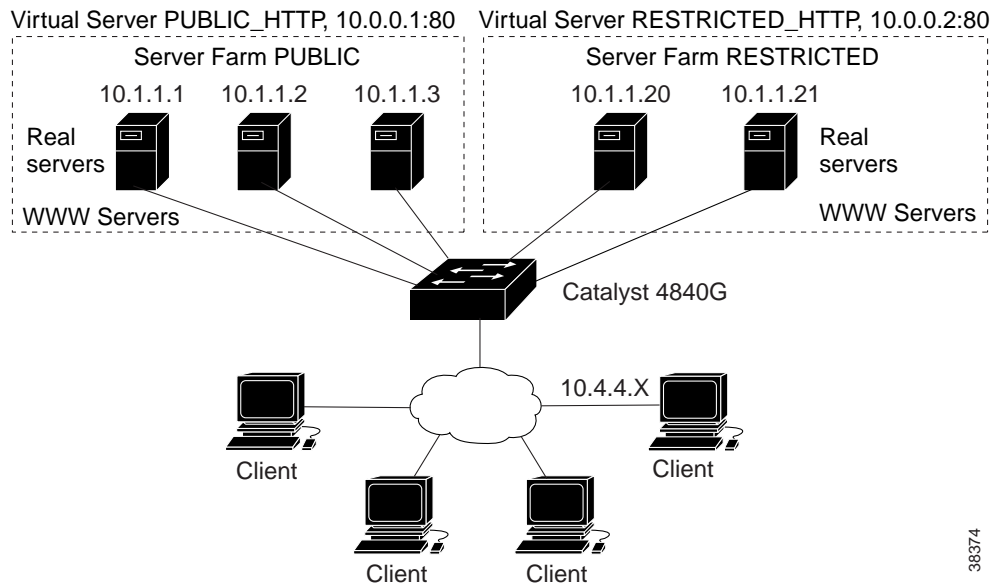
## Sample SLB Device Network

Figure 5-5 shows a sample 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
  - Three real servers in the PUBLIC server farm configured with IP addresses 10.1.1.1, 10.1.1.2, and 10.1.1.3

- Two real servers in the RESTRICTED server farm configured 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. This server is configured with IP address 10.0.0.1 load-balancing TCP connections on the WWW port (80).
  - One configured to allow limited access and named RESTRICTED\_HTTP. This server is configured with IP address 10.0.0.2 load-balancing TCP connections on the WWW port (80) and allows access only to clients from network 10.4.4.X.

Figure 5-5 Sample SLB Device Network



To configure the SLB device network shown in this figure, perform this task, beginning in global configuration mode:

	Command	Purpose
Step 1	SLB-Switch(config)# <b>ip slb serverfarm</b> <i>serverfarm-name</i> SLB-Switch(config-slb-sfarm)#	Adds a server farm definition to the SLB device configuration and initiates server farm configuration mode.
Step 2	SLB-Switch(config-slb-sfarm)# <b>real</b> <i>ip-address</i>	Identifies a real server to the SLB function and initiates real server configuration mode.
Step 3	SLB-Switch(config-slb-real)# <b>inservice</b>	Enables the real server for use by SLB.
Step 4	SLB-Switch(config-slb-real)# <b>exit</b>	Returns to server farm configuration mode.
Step 5	SLB-Switch(config-slb-real)# <b>end</b>	Returns to global configuration mode.
Step 6	SLB-Switch(config)# <b>ip slb vserver</b> <i>virtserver-name</i>	Identifies a virtual server and initiates virtual server configuration mode.
Step 7	SLB-Switch(config-slb-vserver)# <b>virtual</b> <i>ip-address {tcp   udp} port-number</i> [ <i>service service-name</i> ]	Specifies the virtual server IP address, type of connection, port number, and optional service coupling.

	Command	Purpose
Step 8	SLB-Switch(config-slb-vserver)# <b>inservice</b>	Enables the virtual server for use by SLB.
Step 9	SLB-Switch(config-slb-vserver)# <b>client</b> <i>ip-address network-mask</i>	Specifies which clients are allowed to use the virtual server.

## Configuring Server Farms

The following set of commands shows how to configure the server farm PUBLIC and associate its three real servers:

```
SLB-Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
SLB-Switch(config)# ip slb serverfarm PUBLIC
SLB-Switch(config-slb-sfarm)# real 10.1.1.1
SLB-Switch(config-slb-real)# inservice
SLB-Switch(config-slb-real)# exit
SLB-Switch(config-slb-sfarm)# real 10.1.1.2
SLB-Switch(config-slb-real)# inservice
SLB-Switch(config-slb-real)# exit
SLB-Switch(config-slb-sfarm)# real 10.1.1.3
SLB-Switch(config-slb-real)# inservice
SLB-Switch(config-slb-real)# end
```

The following set of commands shows how to configure the server farm RESTRICTED and associate its two real servers:

```
SLB-Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
SLB-Switch(config)# ip slb serverfarm RESTRICTED
SLB-Switch(config-slb-sfarm)# real 10.1.1.20
SLB-Switch(config-slb-real)# inservice
SLB-Switch(config-slb-real)# exit
SLB-Switch(config-slb-sfarm)# real 10.1.1.21
SLB-Switch(config-slb-real)# inservice
SLB-Switch(config-slb-real)# end
SLB-Switch#
```

## Verifying Server Farms

The following command shows how to display the status of server farms PUBLIC and RESTRICTED, and their associated real servers and their status:

```
SLB-Switch# show ip slb real
```

real	server farm	weight	state	conns
10.1.1.1	PUBLIC	8	INSERVICE	0
10.1.1.2	PUBLIC	8	INSERVICE	0
10.1.1.3	PUBLIC	8	INSERVICE	0
10.1.1.20	RESTRICTED	8	INSERVICE	0
10.1.1.21	RESTRICTED	8	INSERVICE	0

```
SLB-Switch#
```

The following command shows how to display the configuration and status of server farms PUBLIC and RESTRICTED:

```
SLB-Switch# show ip slb serverfarm

server farm      predictor    nat    reals    bind id
-----
PUBLIC           ROUNDROBIN  none   3        0
RESTRICTED      ROUNDROBIN  none   2        0
SLB-Switch#
```

## Configuring Virtual Servers

The following set of commands shows how to configure the virtual servers PUBLIC\_HTTP and RESTRICTED\_HTTP:

```
SLB-Switch#
SLB-Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
SLB-Switch(config)# ip slb vserver PUBLIC_HTTP
SLB-Switch(config-slb-vserver)# virtual 10.0.0.1 tcp www
SLB-Switch(config-slb-vserver)# serverfarm PUBLIC
SLB-Switch(config-slb-vserver)# inservice
SLB-Switch(config-slb-vserver)#
16:17:48:
ADD_WILDCARD:
.
(Information Deleted)
.
index = 1
SLB-Switch(config-slb-vserver)# exit
SLB-Switch(config)# ip slb vserver RESTRICTED_HTTP
SLB-Switch(config-slb-vserver)# virtual 10.0.0.2 tcp www
SLB-Switch(config-slb-vserver)# serverfarm RESTRICTED
SLB-Switch(config-slb-vserver)# inservice
SLB-Switch(config-slb-vserver)#
16:20:18:
ADD_WILDCARD:
.
(Information Deleted)
.
index = 1
SLB-Switch(config-slb-vserver)# end
SLB-Switch#
```

## Verifying Virtual Servers

The following command shows how to verify the configuration of the virtual servers PUBLIC\_HTTP and RESTRICTED\_HTTP:

```
SLB-Switch# show ip slb vservers

slb vserver      prot    virtual                                state      conns
-----
PUBLIC_HTTP      TCP     10.0.0.1:80                          INSERVICE  0
RESTRICTED_HTTP TCP     10.0.0.2:80                          INSERVICE  0
SLB-Switch#

SLB-Switch#
```

## Configuring Restricted Clients

The following set of commands shows how to remove the virtual server RESTRICTED\_HTTP from service, allow the restricted client to have access to the virtual server, and then reenable the virtual server:

```
SLB-Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
SLB-Switch(config)# ip slb vserver RESTRICTED_HTTP
SLB-Switch(config-slb-vserver)# no inservice
SLB-Switch(config-slb-vserver)#
16:23:14:
.
(Information Deleted)
.
index = 1
SLB-Switch(config-slb-vserver)# client 10.4.4.0 255.255.255.0
SLB-Switch(config-slb-vserver)# inservice
SLB-Switch(config-slb-vserver)#
16:23:45:
ADD_WILDCARD:
src = 0 - 0
.
(Information Deleted)
.
index = 1
SLB-Switch(config-slb-vserver)# end
SLB-Switch#
```

## Verifying Restricted Clients

The following command shows how to verify restricted client access and status:

```
SLB-Switch# show ip slb conns client 10.4.4.0
```

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

```
SLB-Switch#
```

The following command shows how to display detailed information about restricted client access:

```
SLB-Switch# 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
SLB-Switch#
```

## Verifying SLB Connectivity

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

The following command shows how to display detailed information about the SLB device network status:

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

See the “Monitoring and Maintaining SLB” section on page 5-34 for additional commands used to verify SLB networks and connections.

## Sample SLB Network Configuration

The following command shows how to display the SLB device network shown in Figure 5-5:

```
SLB-Switch# show running-config
Building configuration...

Current configuration:
!
.
(Information Deleted)
.
ip slb probe PROBE2 http
 request method POST url /probe.cgi?all
 header Cookie Monster
 header Authorization Basic U2VtaXN3ZWV0OmNoaXBz
!
ip slb serverfarm PUBLIC
 nat server
  real 10.1.1.1
  inservice
  real 10.1.1.2
  inservice
  real 10.1.1.3
  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
```

```
    inservice
real 10.1.1.21
    inservice
!
ip slb vserver PUBLIC_HTTP
virtual 10.0.0.1 tcp www
serverfarm PUBLIC
no inservice
!
ip slb vserver RESTRICTED_HTTP
virtual 10.0.0.2 tcp www
serverfarm RESTRICTED
no advertise
sticky 60 group 1
idle 1800
client 10.4.4.0 255.255.255.0
synguard 3600000
inservice
!
```

## Optional SLB Configuration Tasks

This section describes a series of optional tasks you can perform to fine-tune the SLB configuration on your Catalyst 4840G SLB switch. It includes the following sections:

- Specifying a Load Balancing Algorithm (Optional), page 5-19
- Configuring HTTP Redirect Load Balancing (Optional), page 5-20
- Specifying a Bind ID (Optional), page 5-21
- Configuring Real Server Attributes (Optional), page 5-21
- Adjusting Virtual Server Values (Optional), page 5-22
- Configuring SLB Dynamic Feedback Protocol (Optional), page 5-23
- Configuring SLB Network Address Translation (Optional), page 5-24
- Configuring HTTP Probe (Optional), page 5-27

### Specifying a Load Balancing Algorithm (Optional)

To determine which real server to use for each new connection request, the SLB feature uses one of two load-balancing algorithms: round-robin (the default) or least-connections. See the “Weighted Round Robin” section on page 5-3 or the “Weighted Least Connections” section on page 5-4 for detailed descriptions of these algorithms.

**Note**

---

You can configure a real server with a weight relative to other real servers in the server farm, by using the **weight (server farm)** real server configuration command.

---

To specify the load-balancing algorithm, perform this task in server farm configuration mode:

Command	Purpose
SLB-Switch(config-slb-sfarm)# <b>predictor</b> [roundrobin   leastconns]	Specifies whether the weighted round-robin algorithm or the weighted least-connections algorithm is to be used to determine how a real server is selected.

This example shows how to configure the weighted least-connections algorithm:

```
SLB-Switch(config)# ip slb serverfarm RESTRICTED
SLB-Switch(config-slb-sfarm)# predictor leastconns
```

## Configuring HTTP Redirect Load Balancing (Optional)

You can use the HTTP redirect load-balancing feature to provide persistence across HTTP and SSL sessions.

The following guidelines apply to HTTP redirect load balancing:

- You can configure only one host name translation string per WSWA virtual server.
- An RSVA IP address can exist in only one server farm.
- If you create a new RSVA virtual server using HTTP redirect code 301, this server may need to be configured with a higher weight to receive an equal share of the load.

This example shows how to configure HTTP redirect load balancing on a server farm:

```
ip slb serverfarm ACME_SF1
 predictor roundrobin http-redirect
 webhost 1 name www.acme.com
 !
 redirect-vserver ACME1_VS
  virtual 40.1.1.1 tcp www
  webhost 1 relocation www1.acme.com/%p 301
  webhost 1 backup www6.acme-inside.com
  ssl https
  inservice
 !
 redirect-vserver ACME2_VS
  virtual 40.1.1.2 tcp www
  webhost 1 relocation www2.acme.com/sales/%p
  webhost 1 backup www1.acme.com 301
  idle 240
  delay 15
  inservice
 !
 real 10.1.1.1
  weight 4
  faildetect numconns 4
  redirect-vserver ACME1_VS
  inservice
 !
```

```

real 10.1.1.2
  weight 4
  reassign 2
  redirect-vserver ACME2_VS
  inservice
!
ip slb vserver ACME_VS
  virtual 40.40.1.40 tcp www
  serverfarm ACME_SF1
  inservice

```

## Specifying a Bind ID (Optional)

The bind ID allows a single real server to be bound to multiple virtual servers and report a different weight, or workload capacity, for each server that it is bound to. The real server is represented as multiple instances of itself, each having a different bind ID. DFP uses the bind ID to identify the instance of the real server that is given a particular weight.

To configure a bind ID on the server farm for use by DFP, perform the following steps in server farm configuration mode:

	Command	Purpose
Step 1	SLB-Switch(config)# <b>ip slb serverfarm</b> <i>serverfarm-name</i> SLB-Switch(config-slb-sfarm)#	Adds a server farm definition to the SLB device configuration and initiates server farm configuration mode.
Step 2	SLB-Switch(config-slb-sfarm)# <b>bindid</b> <i>[bind_id]</i>	Specifies a bind ID on the server farm for use by DFP.

This example shows how to configure a bind ID of 309 on server farm RESTRICTED:

```

SLB-Switch(config)# ip slb serverfarm RESTRICTED
SLB-Switch(config-slb-sfarm)# bindid 309

```

## Configuring Real Server Attributes (Optional)

To configure any of the following real server attributes, perform the following commands in real server configuration mode:

	Command	Purpose
Step 1	SLB-Switch(config-slb-real)# <b>faildetect</b> <b>numconns</b> <i>number-conns</i> [ <b>numclients</b> <i>number-clients</i> ]	Specifies the number of consecutive connection failures and, optionally, the number of unique client connection failures that constitute failure of the real server.
Step 2	SLB-Switch(config-slb-real)# <b>maxconns</b> <i>number-conns</i>	Specifies the maximum number of active connections allowed on the real server at one time.
Step 3	SLB-Switch(config-slb-real)# <b>reassign</b> <i>threshold</i>	Specifies the number of consecutive unanswered SYNs that initiate reassignment of the connection to another real server.

	Command	Purpose
Step 4	SLB-Switch(config-slb-real)# <b>retry</b> <i>retry-value</i>	Specifies the interval, in seconds, that must elapse between detection of a server failure and the next attempt to connect to the failed server.
Step 5	SLB-Switch(config-slb-real)# <b>weight</b> <i>weighting-value</i>	Specifies the real server's workload capacity relative to other servers in the server farm.

## Adjusting Virtual Server Values (Optional)

To change the default settings of the virtual server values, perform the following commands in virtual server configuration mode:

	Command	Purpose
Step 1	SLB-Switch(config)# <b>ip slb vserver</b> <i>virtserver-name</i>	Identifies a virtual server and initiates virtual server configuration mode.
Step 2	SLB-Switch(config-slb-vserver)# <b>no</b> <b>advertise</b>	Omits the virtual server IP address from routing protocol updates.
Step 3	SLB-Switch(config-slb-vserver)# <b>client</b> <i>ip-address network-mask</i>	Specifies which clients are allowed to use the virtual server.
Step 4	SLB-Switch(config-slb-vserver)# <b>delay</b> <i>duration</i>	Specifies the amount of time SLB maintains a TCP connection after the connection has terminated. (The default value is 10 seconds.)
Step 5	SLB-Switch(config-slb-vserver)# <b>idle</b> <i>duration</i>	Specifies the minimum amount of time SLB maintains a connection in the absence of packet activity for the connection. (The default value is 3600 seconds [60 minutes].)
Step 6	SLB-Switch(config-slb-vserver)# <b>sticky</b> <i>duration [group group-id] [netmask netmask]</i>	Specifies that connections from the same client use the same real server, providing that the interval between client connections does not exceed the specified duration (in seconds).
Step 7	SLB-Switch(config-slb-vserver)# <b>synguard</b> <i>syn-count interval</i>	Specifies the rate of TCP SYNs handled by a virtual server in order to prevent a SYN flood denial of service attack.
Step 8	SLB-Switch(config-slb-vserver)# <b>inservice</b>	Enables the virtual server for use by SLB.

This example shows how to remove the virtual server RESTRICTED\_HTTP from service and then allow the restricted client in Figure 5-5 to have access to the virtual server:

```
SLB-Switch(config)# ip slb vserver RESTRICTED_HTTP
SLB-Switch(config-slb-vserver)# no inservice
SLB-Switch(config-slb-vserver)#
16:23:14:
.
.
(Information Deleted)
.
index = 1
SLB-Switch(config-slb-vserver)# client 10.4.4.0 255.255.255.0
```

By default, virtual server addresses are advertised. That is, static routes to the Null0 interface are installed for the virtual server addresses. To advertise these static routes using the routing protocol, you must configure redistribution of static routes for the routing protocol. To prevent the installation of a static route, use the **no** form of the **advertise** command:

```
SLB-Switch(config-slb-vserver) # no advertise
```

This example shows how to configure the delay timer to 20 seconds after the termination of the TCP connection to the virtual server:

```
SLB-Switch(config-slb-vserver) # delay 20
```

The following command configures 180 seconds (3 minutes) as the idle time during which the SLB device maintains connectivity to the virtual server in the absence of packet activity:

```
SLB-Switch(config-slb-vserver) # idle 180
```

This example shows how to configure 60 seconds as the time that connections from the same client use the same real server:

```
SLB-Switch(config-slb-vserver) # sticky 60 group 1
```

This example shows how to configure the rate of TCP SYNs handled by the virtual server to 3600000:

```
SLB-Switch(config-slb-vserver) # synguard 3600000
```

## Configuring SLB Dynamic Feedback Protocol (Optional)

SLB Dynamic Feedback Protocol (DFP) allows host agents in load-balanced environments to dynamically report the change in status of the host systems providing a virtual service. The status reported is a relative weight that specifies a host server's capacity to perform work.

To configure SLB DFP, perform the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	SLB-Switch(config)# <b>ip slb dfp</b> [ <b>password password</b> [ <b>timeout</b> ]]	Configures DFP and optionally sets a password and initiates DFP configuration mode.
Step 2	SLB-Switch(config-slb- <b>dfp</b> )# <b>agent ip_address port</b> [ <b>timeout</b> [ <b>retry_count</b> [ <b>retry_interval</b> ]]]	Configures a DFP agent. See the <b>agent</b> command for more information.
Step 3	SLB-Switch(config-slb- <b>dfp</b> )# <b>manager [port]</b>	Assigns a DFP manager to the specified port. See the <b>manager</b> command for more information

This example shows how to make the following configurations:

- Set the DFP password to Cookies and the timeout to 360 seconds.
- Specify that the SLB manager connects to the agent at IP address 10.1.1.1 and port 2221.

```
SLB-Switch(config)# ip slb dfp password Cookies 360
SLB-Switch(config-slb-dfp)# agent 10.1.1.1 2221
```

- On the DFP agent, assign the DFP manager to port 2345:

```
SLB-Switch(config)# ip slb dfp
SLB-Switch(config-slb-dfp)# manager 2345
```

## Configuring SLB Network Address Translation (Optional)

To configure SLB NAT for client mode, perform the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	SLB-Switch(config)# <b>ip slb natpool</b> <i>pool-name</i> <i>start-ip end-ip [netmask mask   leading_1_bits]</i>	Specifies the client address pool.
Step 2	SLB-Switch(config-slb-sfarm)# <b>nat</b> { <b>server</b>   <b>client</b> <i>pool-name</i> }	Specifies which NAT mode to use.

This example shows how to configure a NAT server on server farm PUBLIC:

```
SLB-Switch(config)# ip slb serverfarm PUBLIC
SLB-Switch(config-slb-sfarm)# nat client
```

To configure SLB NAT client mode for a specific server farm, perform the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	SLB-Switch(config)# <b>ip slb natpool</b> <i>pool-name start-ip end-ip [netmask mask  </i> <i>leading_1_bits]</i>	Specifies the client address pool.
Step 2	SLB-Switch(config)# <b>ip slb serverfarm</b> <i>serverfarm-name</i>	Adds a server farm definition to the SLB device configuration and initiates server farm configuration mode.
Step 3	SLB-Switch(config-slb-sfarm)# <b>nat</b> { <b>server</b>   <b>client</b> <i>pool-name</i> }	Specifies which NAT mode to use.
Step 4	SLB-Switch(config-slb-sfarm)# <b>real</b> <i>ip-address [port_number]</i>	Identifies a real server to the SLB function and initiates real server configuration mode.
Step 5	SLB-Switch(config-slb-real)# <b>inservice</b>	Enables the real server for use by SLB.
Step 6	SLB-Switch(config-slb-real)# <b>exit</b>	Returns to server farm configuration mode.
Step 7	SLB-Switch(config-slb-real)# <b>end</b>	Returns to global configuration mode.
Step 8	SLB-Switch(config)# <b>ip slb vserver</b> <i>virtserver-name</i>	Identifies a virtual server and initiates virtual server configuration mode.

This example shows how to configure NAT clients on server farm FARM2:

```
SLB-Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
SLB-Switch(config)# ip slb natpool Web-clients 128.3.0.1 128.3.0.254 netmask 255.255.0.0
SLB-Switch(config)# ip slb serverfarm FARM2
SLB-Switch(config-slb-sfarm)# nat client Web-clients
SLB-Switch(config-slb-sfarm)# real 10.3.1.1
SLB-Switch(config-slb-real)# inservice
SLB-Switch(config-slb-real)# exit
SLB-Switch(config-slb-sfarm)# real 10.4.1.1 8080
SLB-Switch(config-slb-real)# inservice
SLB-Switch(config-slb-real)# exit
SLB-Switch(config-slb-sfarm)# real 10.4.1.1 8081
SLB-Switch(config-slb-real)# inservice
SLB-Switch(config-slb-real)# exit
SLB-Switch(config-slb-sfarm)# real 10.4.1.1 8082
SLB-Switch(config-slb-real)# inservice
SLB-Switch(config-slb-real)# end
SLB-Switch#
00:27:00: %SYS-5-CONFIG_I: Configured from console by console
SLB-Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
SLB-Switch(config)# ip slb vserver HTTP2
SLB-Switch(config-slb-vserver)# virtual 128.2.0.1 tcp www
SLB-Switch(config-slb-vserver)# serverfarm FARM2
SLB-Switch(config-slb-vserver)# inservice
SLB-Switch(config-slb-vserver)#
00:28:21:
ADD_WILDCARD:

(Information Deleted)

SLB-Switch# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
SLB-Switch#
```

This example shows the final configuration after a NAT client pool was configured using the previous task:

```
SLB-Switch# show running-config
Building configuration...

(Information Deleted)

!
ip slb natpool WEB-CLIENTS 128.3.0.1 128.3.0.254 netmask 255.255.0.0 entries 800
0 13851890
ip slb serverfarm FARM2
nat server
nat client WEB-CLIENTS
real 10.3.1.1
faildetect numconns 8 numclients 2
inservice
real 10.4.1.1 8080
faildetect numconns 8 numclients 2
inservice
real 10.4.1.1 8081
faildetect numconns 8 numclients 2
inservice
real 10.4.1.1 8082
```

```

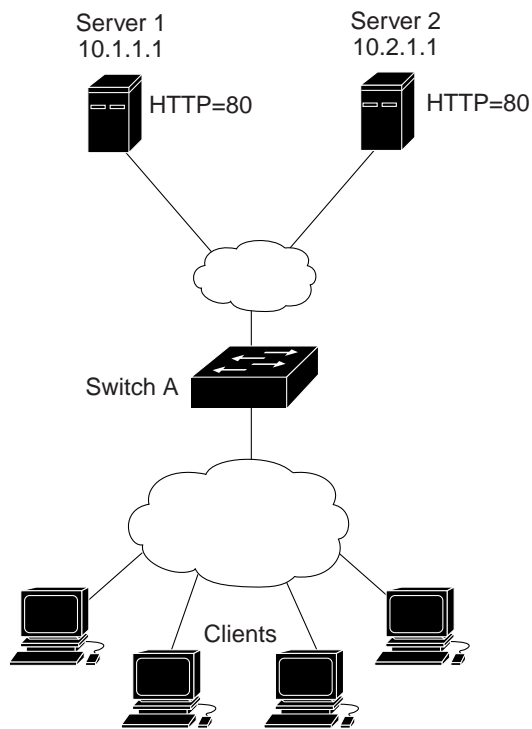
    faildetect numconns 8 numclients 2
    inservice
!
ip slb vserver HTTP2
  virtual 128.2.0.1 tcp www
  serverfarm FARM2
  inservice
!

```

## Sample NAT Configuration

Figure 5-6 shows a sample SLB NAT configuration topology.

**Figure 5-6** Sample NAT Topology



38200

The topology in Figure 5-6 has two Web servers running a single HTTP server application listening on port 80.

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

The configuration statements for Switch A are:

```

ip slb serverfarm FARM1
! Translate server addresses
  nat server
! Server 1 port 80
  real 10.1.1.1
  inservice
! Server 2 port 80
  real 10.2.1.1
  inservice
!

```

```

ip slb vserver HTTP1
! Handle HTTP (port 80) requests
virtual 128.1.0.1 tcp www
serverfarm FARM1
inservice

```

## Configuring HTTP Probe (Optional)

An HTTP probe allows you to monitor the applications that are server load balanced. With frequent probes you can verify the operation of the application and connectivity to the application.

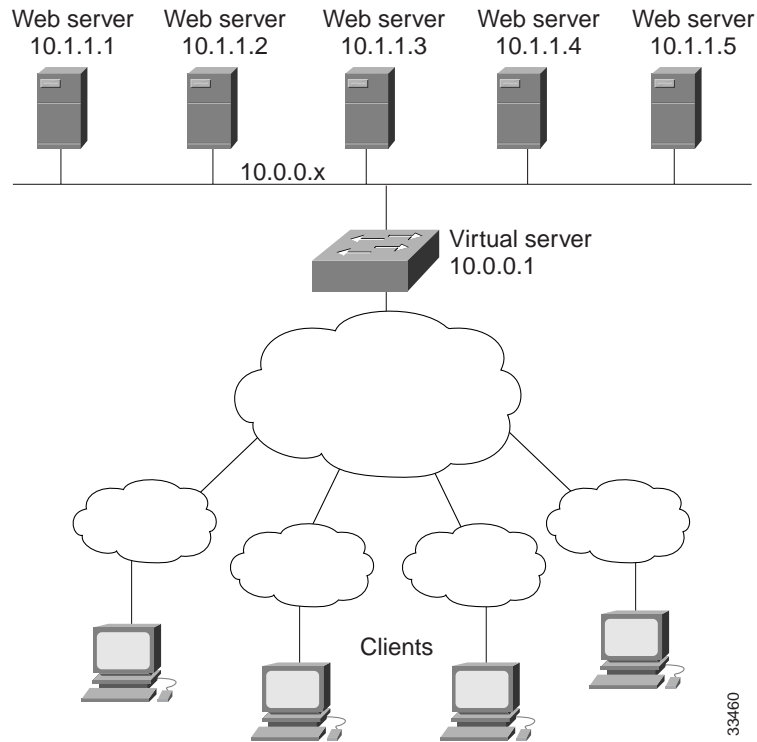
The basic requirement of an HTTP probe is to determine the real server status, by issuing an HTTP GET against each real server in a server farm. For a detailed description, see the “TCP Session Reassignment” section on page 5-9.

To configure an HTTP probe, perform the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	SLB-Switch(config)# <b>ip slb probe name http</b>	Specifies the SLB probe name and changes to HTTP configuration submenu.
Step 2	SLB-Switch(config-slb-probe)# <b>request method {get   post   head   name name [url]}</b>	Configures the method used to perform the request to the server.
Step 3	SLB-Switch(config-slb-probe)# <b>expect [status number]</b>	Configures the expected HTTP status code.
Step 4	SLB-Switch(config-slb-probe)# <b>interval seconds</b>	Configures the HTTP probe transmit timers.
Step 5	SLB-Switch(config-slb-probe)# <b>header {field-name}</b>	Configures basic authentication values for the HTTP probe.
Step 6	SLB-Switch(config-slb-probe)# <b>credentials {username [password]}</b>	Configures basic authentication values for the HTTP probe.
Step 7	SLB-Switch(config-slb-real)# <b>exit</b>	Returns to global configuration mode.
Step 8	SLB-Switch(config)# <b>ip slb serverfarm serverfarm-name</b>	Changes to server farm configuration mode.
Step 9	SLB-Switch(config-slb-sfarm)# <b>probe probe-name</b>	Specifies an HTTP probe on the server farm.

Figure 5-7 shows a sample configuration with IOS SLB real server connections configured as part of a server farm. The example focuses on using ping and HTTP probes to monitor applications being server load balanced.

Figure 5-7 Sample Ping and HTTP Probe Topology



The topology shown in Figure 5-7 is a heterogeneous server farm servicing a single 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
SLB-Switch(config)# ip slb probe PROBE1 ping
! Configure probe to receive responses from IP address 13.13.13.13
SLB-Switch(config-slb-probe)# address 13.13.13.13
! Configure unanswered ping threshold to 16
SLB-Switch(config-slb-probe)# faildetect 16
! Configure ping probe timer interval to transmit every 11 seconds
SLB-Switch(config-slb-probe)# interval 11
! Configure HTTP probe PROBE2
SLB-Switch(config-slb-probe)# ip slb probe PROBE2 http
! Configure request method as POST, set URL as /probe.cgi?all
SLB-Switch(config-slb-probe)# request method post url /probe.cgi?all
! Configure header Cookie Monster
SLB-Switch(config-slb-probe)# header name Cookie Monster
! Configure basic authentication username and password
SLB-Switch(config-slb-probe)# credentials Semisweet chips
! Exit to global configuration mode
SLB-Switch(config-slb-probe)# exit
! Enter IOS SLB server farm configuration mode for server farm PUBLIC
SLB-Switch(config)# ip slb serverfarm PUBLIC
! Configure NAT server and real servers on the server farm
SLB-Switch(config-slb-sfarm)# nat server
SLB-Switch(config-slb-sfarm)# real 10.1.1.1
SLB-Switch(config-slb-sfarm)# inservice
SLB-Switch(config-slb-sfarm)# real 10.1.1.2
SLB-Switch(config-slb-sfarm)# inservice
SLB-Switch(config-slb-sfarm)# real 10.1.1.3
SLB-Switch(config-slb-sfarm)# inservice

```

```

SLB-Switch(config-slb-sfarm)# real 10.1.1.4
SLB-Switch(config-slb-sfarm)# inservice
SLB-Switch(config-slb-sfarm)# real 10.1.1.5
SLB-Switch(config-slb-sfarm)# inservice
! Configure ping probe on the server farm
SLB-Switch(config-slb-sfarm)# probe PROBE1
! Configure HTTP probe on the server farm
SLB-Switch(config-slb-sfarm)# probe PROBE2
SLB-Switch(config-slb-sfarm)# end
!

```

## Required URL-Based SLB Configuration Tasks

This section describes the steps necessary to configure basic URL-based SLB on the Catalyst 4840G SLB switch.

Configuring URL-based SLB involves configuring URL maps, identifying server farms, configuring policies, and configuring policy-based virtual servers. See the following sections for required configuration tasks for the URL-based SLB feature:

- Configuring URL Maps, page 5-31
- Verifying URL Maps, page 5-31
- Configuring Policies, page 5-31
- Verifying Policies, page 5-31
- Configuring Policy-Based Virtual Servers, page 5-32
- Verifying Policy-Based Virtual Servers, page 5-32
- Sample URL-Based SLB Network Configuration, page 5-32

For an example of configuring server farms, see the “Configuring Server Farms” section on page 5-15.

To configure the URL-based SLB network shown in Figure 5-2, perform the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	SLB-Switch(config)# <b>ip slb serverfarm</b> <i>serverfarm-name</i>	Adds a server farm definition to the SLB device configuration and initiates server farm configuration mode.
Step 2	SLB-Switch(config-slb-sfarm)# <b>real</b> <i>ip-address</i>	Identifies a real server to the SLB function and initiates real server configuration mode.
Step 3	SLB-Switch(config-slb-real)# <b>inservice</b>	Enables the real server for use by SLB.
Step 4	SLB-Switch(config-slb-real)# <b>exit</b>	Returns to server farm configuration mode.
Step 5	SLB-Switch(config-slb-real)# <b>end</b>	Returns to global configuration mode.
Step 6	SLB-Switch# <b>ip slb map</b> <i>url-map-name</i>	Configures a URL map in URL map submenu. Use the <b>no</b> form of this command to restore the defaults.
Step 7	SLB-Switch(ip-slb-map)# <b>match protocol</b> <b>http url</b> <i>url1</i>	Configures multiple URLs into a URL map in URL map submenu. Regular expressions for <i>url1</i> and <i>url2</i> are based on UNIX filename specifications. A maximum of four URLs can be configured to a single map.

	Command	Purpose
Step 8	SLB-Switch(ip-slb-map)# <b>exit</b>	Returns to global configuration mode.
Step 9	SLB-Switch (config)# <b>ip slb policy</b> <i>policy-name</i>	Configures the IOS Content Switching policy attributes, and then associates the attributes with a virtual server. Use the <b>no</b> form of this command to restore the defaults.
Step 10	SLB-Switch (config-ip-slb)# <b>url-map</b> <i>url-map-name</i>	Configures a list of URLs with a policy. Use the <b>no</b> form of this command to restore the defaults.
Step 11	SLB-Switch (config-ip-slb)# <b>serverfarm</b> <i>serverfarm_name</i>	Configures the server farm serving a particular load-balancing policy. Only one server farm can be configured per policy. Use the <b>no</b> form of this command to restore the defaults.
Step 12	SLB-Switch(config-ip-slb)# <b>exit</b>	Returns to global configuration mode.
Step 13	SLB-Switch(config)# <b>ip slb vserver</b> <i>virtserver-name</i>	Identifies a policy-based virtual server and initiates virtual server configuration mode.
Step 14	SLB-Switch(config-slb-vserver)# <b>virtual</b> <i>ip-address</i> { <b>tcp</b>   <b>udp</b> } <i>port-number</i> [ <b>service</b> <i>service-name</i> ]	Specifies the policy-based virtual server IP address, type of connection, port number, and optional service coupling.
Step 15	SLB-Switch(config-slb-vserver)# <b>slb-policy</b> <i>policy_content1</i>	Specifies the policies associated with a particular policy-based virtual server.
Step 16	SLB-Switch(config-slb-vserver)# <b>serverfarm</b> <i>serverfarm_name</i>	Configures the server farm serving a particular virtual server. Use the <b>no</b> form of this command to restore the defaults.
Step 17	SLB-Switch(config-slb-vserver)# <b>inservice</b>	Enables the policy-based virtual server for use by SLB.
Step 18	SLB-Switch (config)> <b>ip slb vserver</b> <i>virtserver-name</i>	Identifies the virtual server. Use the <b>no</b> form of this command to restore the defaults.
Step 19	SLB-Switch (config-ip-slb)> <b>timeout</b> <i>duration</i>	Configures how long to wait before a TCP connection is considered to have failed. Use the <b>no</b> form of this command to restore the defaults.
Step 20	SLB-Switch (config-ip-slb)> <b>idle</b> <i>duration</i>	Configures the amount of time IOS Content Switching maintains connection information in the absence of packet activity for a connection. Use the <b>no</b> form of this command to restore the defaults.
Step 21	SLB-Switch (config-ip-slb)> <b>retransmit</b> <i>max-count</i> <i>tbd</i>	Configures the maximum number of retransmissions per TCP packet for a virtual server. Use the <b>no</b> form of this command to restore the defaults.
Step 22	SLB-Switch (config-ip-slb)> <b>delay</b> <i>duration</i> <i>not supported by Laminar</i>	Configures the amount of time IOS Content Switching maintains a TCP connection context after a connection has terminated. Use the <b>no</b> form of this command to restore the defaults.
Step 23	SLB-Switch (config-ip-slb)> <b>synguard</b> <i>syn-count</i> [ <i>interval</i> ]	Ensures that only <i>syn-count</i> number of SYNs are allowed to remain unanswered over <i>interval</i> .
Step 24	SLB-Switch (config-ip-slb)> <b>show ip slb</b> <b>tcp</b> [ <b>vserver</b> <i>vserver-name</i> ]	Displays TCP information for virtual servers defined for Content Switching.
Step 25	SLB-Switch(config-ip-slb)# <b>exit</b>	Returns to global configuration mode.

## Configuring URL Maps

This example shows how to group URLs and associate them with a Content Switching policy:

```
SLB-Switch(config)#ip slb map m1
SLB-Switch(config-slb-map)#match protocol http url /index.html
SLB-Switch(config-slb-map)#match protocol http url /stocks/cisco/
SLB-Switch(config-slb-map)#match protocol http url *gif
SLB-Switch(config-slb-map)#match protocol http url /st*
SLB-Switch(config-slb-map)#exit
```

## Verifying URL Maps

This example shows how to display URL maps associated with a Content Switching policy:

```
SLB-Switch#show ip slb map
URL map M1 rules:
/
SLB-Switch#
```

## Configuring Policies

This example shows how to configure URL-based SLB policies:

```
SLB-Switch(config)#ip slb policy policy_content
SLB-Switch(config-slb-policy)#serverfarm new_serverfarm
SLB-Switch(config-slb-policy)#url-map url_map_1
SLB-Switch(config-slb-policy)#exit
SLB-Switch(config)
```

## Verifying Policies

This example shows how to display the policies associated with a Content Switching policy:

```
SLB-Switch#show ip slb policy
URL policy:          P1
  url map:           M1
  serverfarm:       SF1

URL policy:          POLICY_CONTENT
  url map:           URL_MAP_1
  serverfarm:       NEW_SERVERFARM

URL policy:          SHOW
  url map:
  serverfarm:
SLB-Switch#
```

## Configuring Policy-Based Virtual Servers

Configuring a policy-based virtual server involves configuring the attributes of the virtual server and associating it with one or more URL-based policies and a default server farm.

This example shows how to configure a policy-based virtual server named *CONTENT\_AWARE*:

```
SLB-Switch(config)# ip slb vserver CONTENT_AWARE
SLB-Switch(config-slb-vserver)# virtual 10.0.0.1 tcp www
SLB-Switch(config-slb-vserver)# serverfarm PUBLIC
SLB-Switch(config-slb-vserver)# slb-policy policy_content1
SLB-Switch(config-slb-vserver)# slb-policy policy_content2
SLB-Switch(config-slb-vserver)# slb-policy policy_content3
SLB-Switch(config-slb-vserver)# inservice
SLB-Switch(config-slb-vserver)#
16:17:48:
ADD_WILDCARD:
.
```

This example shows how to configure a policy-based virtual server named *content\_aware*:

```
SLB-Switch(config-ip-slb)> ip slb vserver vs_content_aware
SLB-Switch(config-ip-slb)> slb-policy policy_content
SLB-Switch(config-ip-slb)> serverfarm default_serverfarm
SLB-Switch(config-ip-slb)> exit
```

## Verifying Policy-Based Virtual Servers

This example shows how to verify the configuration of the policy-based virtual servers:

```
SLB-Switch# show ip slb vservers
```

slb vserver	prot	virtual	state	conns
PUBLIC_HTTP	TCP	10.0.0.1:80	INSERVICE	0
RESTRICTED_HTTP	TCP	10.0.0.2:80	INSERVICE	0
CONTENT_AWARE	TCP	10.0.0.2:80	INSERVICE	0

```
SLB-Switch#
SLB-Switch#
```

## Sample URL-Based SLB Network Configuration

This example shows how to display the URL-based SLB network shown in Figure 5-2:

```
SLB-Switch# show running-config
ip slb serverfarm IMAGES
  predictor leastconns
  ...
  real 10.1.1.1
  ...
  inservice
  real 10.1.1.2
  ...
  inservice
!
ip slb serverfarm STOCKS
  predictor leastconns
  ...
```

```
    real 10.1.2.1
  ...
  inservice
  real 10.1.2.2
  ...
  inservice
  !
ip slb serverfarm ABC
  predictor leastconns
  ...
  real 10.1.3.1
  ...
  inservice
  !
ip slb map image_url
  match protocol http url /images/ method GET

ip slb map stock_url
  match protocol http url /stocks/ method GET

ip slb policy image_policy
  url_map image_url
  serverfarm IMAGES

ip slb policy stock_policy
  url_map stock_policy
  serverfarm STOCKS

ip slb vserver PUBLIC_ABC
  virtual 172.20.1.10 tcp www
  ...
  slb-policy image_policy
  slb_policy stock_policy
  serverfarm ABC
  inservice
```

## Optional URL-Based SLB Configuration Tasks

This section describes the steps you can take to fine-tune the URL-based SLB configuration on your Catalyst 4840G SLB switch. It includes the following sections:

- Configuring TCP Parameters (Optional), page 5-33
- Configuring HTTP 1.1 (Optional), page 5-34

### Configuring TCP Parameters (Optional)

This example shows how configure TCP parameters for virtual servers:

```
SLB-Switch (config)> ip slb vserver barnett
SLB-Switch(config-slb-vserve)#idle 255
SLB-Switch(config-slb-vserve)#exit
SLB-Switch (config)> timeout 10
SLB-Switch (config)> exit
```

## Configuring HTTP 1.1 (Optional)

This example shows how configure HTTP 1.1:

```
SLB-Switch (config)> ip slb http11 enable
SLB-Switch (config)> exit
```

## Monitoring and Maintaining SLB

You can display runtime information about SLB using these commands in EXEC mode:

Command	Purpose
SLB-Switch# <b>show ip slb conns</b> [vserver virtserver-name] [client ip-address] [detail]	Displays all connections handled by SLB or, optionally, only those connections associated with a particular virtual server or client.
SLB-Switch# <b>show ip slb dfp</b> [agent ip_address port   manager [ip_addr]   detail   weights]	Displays information about DFP, DFP agents, DFP managers, and the weights assigned to real servers.
SLB-Switch# <b>show ip slb probe</b> [name probe_name] [detail]	Displays information about SLB HTTP probes configured on real servers.
SLB-Switch# <b>show ip slb reals</b> [vserver virtserver-name] [detail]	Displays information about the real servers defined to SLB.
SLB-Switch# <b>show ip slb serverfarms</b> [name serverfarm-name] [detail]	Displays information about the server farms defined to SLB.
SLB-Switch# <b>show ip slb stats</b>	Displays SLB statistics.
SLB-Switch# <b>show ip slb sticky</b> [client ip-address]	Displays information about the sticky connections defined to SLB.
SLB-Switch# <b>show ip slb vservers</b> [name virtserver-name] [detail]	Displays information about the virtual servers defined to SLB.
SLB-Switch# <b>show ip slb probe</b> [name probe_name] [detail]	Displays information about the HTTP probe defined to SLB.
SLB-Switch# <b>show ip slb replicate</b>	Displays information about the SLB replication configuration.