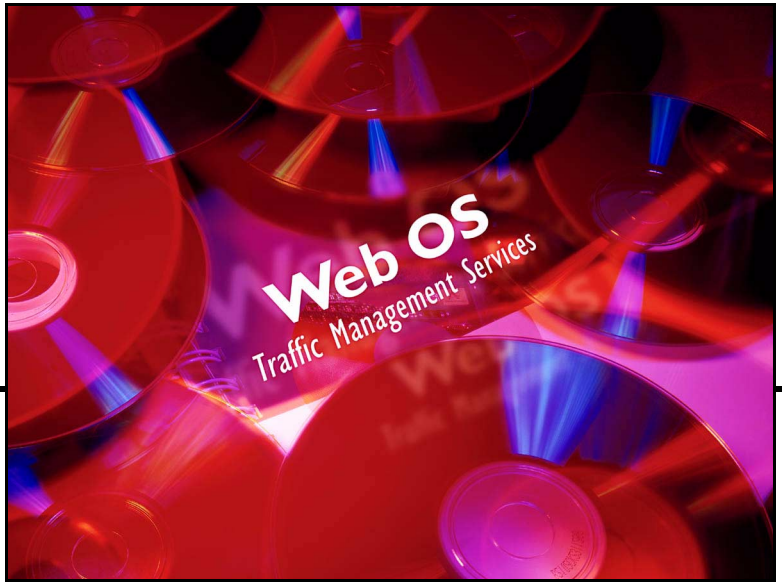
Web OS Switch Software



7.0 Application Guide

Part Number: 050108, Revision A, July 2000

Alteon WebSystems
Web Speed for e-Business

50 Great Oaks Boulevard
San Jose, California 95119
408-360-5500 Main
408-360-5501 Fax
www.alteonwebsystems.com

Copyright 2000 Alteon WebSystems, Inc., 50 Great Oaks Boulevard, San Jose, California 95119, USA. All rights reserved. Part Number: 050108, Revision A.

This document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this document may be reproduced in any form by any means without prior written authorization of Alteon WebSystems, Inc. Documentation is provided “as is” without warranty of any kind, either express or implied, including any kind of implied or express warranty of non-infringement or the implied warranties of merchantability or fitness for a particular purpose.

U.S. Government End Users: This document is provided with a “commercial item” as defined by FAR 2.101 (Oct 1995) and contains “commercial technical data” and “commercial software documentation” as those terms are used in FAR 12.211-12.212 (Oct 1995). Government End Users are authorized to use this documentation only in accordance with those rights and restrictions set forth herein, consistent with FAR 12.211- 12.212 (Oct 1995), DFARS 227.7202 (JUN 1995) and DFARS 252.227-7015 (Nov 1995).

Alteon WebSystems, Inc. reserves the right to change any products described herein at any time, and without notice. Alteon WebSystems, Inc. assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by Alteon WebSystems, Inc. The use and purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of Alteon WebSystems, Inc.

Web OS, Alteon 700 Series, Alteon 708, and Alteon 714 are trademarks of Alteon WebSystems, Inc. in the United States and certain other countries. Cisco® and EtherChannel® are registered trademarks of Cisco Systems, Inc. in the United States and certain other countries. Any other trademarks appearing in this manual are owned by their respective companies.

Contents

Preface 5
Who Should Use This Book 5
What You'll Find in This Book 5
Typographic Conventions 6
Contacting Alteon WebSystems 7

Server Load Balancing 9
Overview 9
- Benefits 9
- Identifying Your Network Needs 10
- How SLB Works 10

Network Topology Considerations 12
SLB Implementation 14
- Web Hosting Configuration 14

Additional SLB Options 20
- IP Address Ranges Using imask 20
- Health Checks for Real Servers 20
- Metrics for Real Server Groups 21
- Weights for Real Servers 23
- Connection Time-outs for Real Servers 23
- Maximum Connections for Real Servers 23
- Backup/Overflow Servers 24
- Mapping Virtual Ports to Real Ports 24
- Proxy IP Addresses for Complex SLB Networks 25

Direct Client Access to Real Servers 28
- Direct Access Mode 28
- Multiple IP Addresses on the Server 28
- Proxy IP Addresses 28
- Port Mapping 29
- Management Network 30

Application Redirection 31
Web-Cache Redirection Example 31
Web-Cache Redirection Environment 32
Example Configuration for the Web-Cache Solution 33
IP Proxy Addresses for Transparent Proxies or Complex Networks 38
Excluding Non-Cacheable Sites 39
Defining IP Address Ranges for Filters 40

Index 41

Preface

This *Application Guide* describes how to configure and use the Web OS 7.0 software included in the Alteon WebSystems family of switches.

For documentation on installing the switches physically, see the *Hardware Installation Guide* for your particular switch model.

Who Should Use This Book

This *Application Guide* is intended for network installers and system administrators engaged in configuring and maintaining a network. It assumes that you are familiar with Ethernet concepts, IP addressing, the IEEE 802.1d Spanning-Tree Protocol, and SNMP configuration parameters.

What You'll Find in This Book

The chapters in this book will help you plan, implement, and administer the use of Web OS 7.0 software features. Where possible, each chapter provides a conceptual overview of a specific Web OS feature or functional area, usage examples, and, configuration instructions for implementing the feature(s) on an Alteon WebSystems switch.

Typographic Conventions

The following table describes the typographic styles used in this book.

Table 1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	This type is used for names of commands, files, and directories used within the text.	View the `readme.txt` file.
	It also depicts on-screen computer output and prompts.	`Main#`
`AaBbCc123`	This bold type appears in command examples. It shows text that must be typed in exactly as shown.	`Main#` **`sys`**
AaBbCc123	This italicized type appears in command examples as a parameter placeholder. Replace the indicated text with the appropriate real name or value when using the command.	To establish a Telnet session, enter: `host#` **`telnet`** *IP-address*
	This also shows book titles, special terms, or words to be emphasized.	Read your *User's Guide* thoroughly.
[ ]	Command items shown inside brackets are optional and can be used or excluded as the situation demands. Do not type the brackets.	`host#` **`ls`** [**`-a`**]

Contacting Alteon WebSystems

Use the following information to access Alteon WebSystems support and sales.

- URL for Alteon WebSystems Online:

  `http://www.alteonwebsystems.com`

  This website includes product information, software updates, release notes, and white papers. The website also includes access to Alteon WebSystems Customer Support for accounts under warranty or that are covered by a maintenance contract.

- E-mail access:

  `support@alteonwebsystems.com`

  E-mail access to Alteon WebSystems Customer Support is available to accounts that are under warranty or covered by a maintenance contract.

- Telephone access to Alteon WebSystems Customer Support:

  1-888-Alteon0 (or 1-888-258-3660)
  1-408-360-5695

  Telephone access to Alteon WebSystems Customer Support is available to accounts that are under warranty or covered by a maintenance contract. Normal business hours are 8 a.m. to 6 p.m. Pacific Standard Time.

- Telephone access to Alteon WebSystems Sales:

  1-888-Alteon2 (or 1-888-258-3662), and press 2 for Sales
  1-408-360-5600, and press 2 for Sales

  Telephone access is available for information regarding product sales and upgrades.

CHAPTER 1

Server Load Balancing

This chapter describes how to configure and use the optional Layer 4 software for Server Load Balancing (SLB). For information on activating this optional software, see your *Web OS 7.0 Command Reference*.

Overview

Benefits

Server Load Balancing benefits your network in a number of ways:

- Increased efficiency for server utilization and network bandwidth

  With SLB, your Alteon 700 series switch is aware of the shared services provided by your server pool. The switch can then balance user session traffic among the available servers. For even greater control, traffic is distributed according to a variety of user-selectable rules.

  By helping to eliminate server over-utilization, important session traffic gets through more easily, reducing user competition for connections on overworked servers.

- Increased reliability of services to users

  If any server in a server pool fails, the remaining servers continue to provide access to vital applications and data. The failed server can be brought back up without interrupting access to services.

- Increased scalability of services

  As users are added and the server pool's capabilities are saturated, new servers can be added to the pool transparently.

Identifying Your Network Needs

SLB may be the right option for addressing these vital network concerns:

- A single server no longer meets the demand for its particular application.
- The connection from your LAN to your server overloads the server's capacity.
- Your NT and UNIX servers hold critical application data and must remain available even in the event of a server failure.
- Your website is being used as a way to do business and for taking orders from customers. It must not become overloaded or unavailable.
- You want to use multiple servers or hot-standby servers for maximum server uptime.
- You must be able to scale your applications to meet client and LAN request capacity.
- You can't afford to continue using an inferior load balancing technique such as DNS round robin, or a software-only system.

How SLB Works

In an average network that employs multiple servers without server load balancing, each server usually specializes in providing one or two unique services. If one of these servers provides access to applications or data that is in high demand, it can become over-utilized. Placing this kind of strain on a server can decrease the performance of the entire network as user requests are rejected by the server and then resubmitted by the user stations. Ironically, over-utilization of key servers often happens in networks where other servers are actually under-utilized.

The solution to getting the most from your servers is SLB, an optional feature on Alteon WebSystems Web switches. With this software feature, the switch is aware of the services provided by each server, and can direct user session traffic to an appropriate server based on a variety of load balancing algorithms.

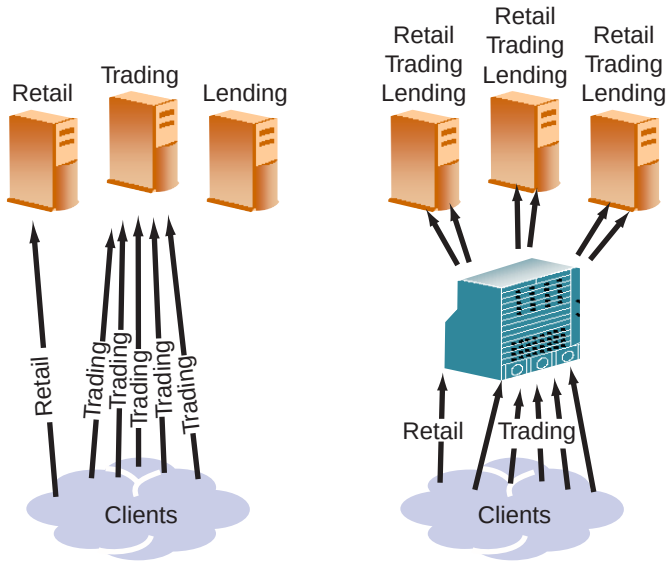


Figure 1 Traditional vs. SLB network configurations

To provide load balancing for any particular type of service, each server in the pool must have access to identical content, either directly (duplicated on each server) or through a back-end network (mounting the same file system or database server).

The Web switch, with SLB software, acts as a front-end to the servers, interpreting user session requests and distributing them among the available servers. To accomplish this, the switch is configured to act as a virtual server and is given a virtual IP address (or range of addresses) for each collection of services it will distribute. Depending on your switch model, there can be as many as 1024 virtual servers on the switch, each distributing up to eight different services (up to a total of 4000 services).

Each virtual server is assigned a list of the real IP addresses (or range of addresses) of the real servers in the pool where its services reside. When the user stations request connections to a service, they will communicate with a virtual server on the switch. When the switch receives the request, it binds the session to the real IP address of the best available real server, and remaps the fields in each frame from virtual addresses to real addresses.

Network Topology Considerations

When deploying SLB, there are a few key aspects to consider:

- In standard SLB, all client requests to a virtual IP address and all responses from the real servers must pass through the switch.

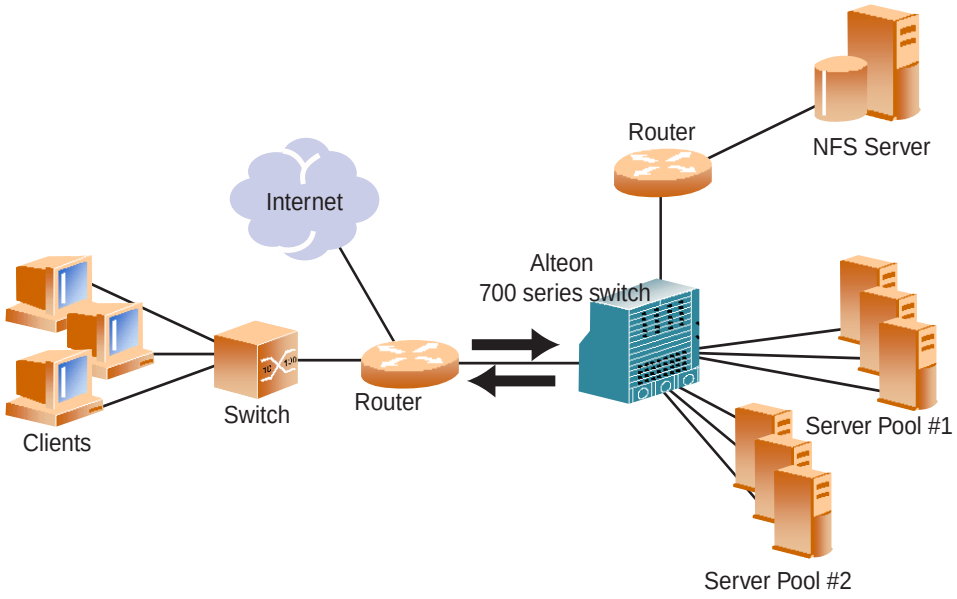


Figure 2 SLB Client/Server traffic routing

- Identical content must be available to each server in the same pool. Either of these methods can be used:
  - Static applications and data are duplicated on each real server in the pool.
  - Each real server in the pool has access to the same data through use of a shared file system or back-end database server.
- Some services require that a series of client requests goes to the same real server so that session-specific state data can be retained between connections.

  Services of this nature include Web search results, multi-page forms that the user fills in, or custom Web-based applications typically created using `cgi-bin` scripts. Connections for these types of services must be configured as "persistent", or must use the `min-misses` or `hash` metrics (see "Metrics for Real Server Groups" on page 21).

- Clients and servers can be connected through the same switch port. Each port in use on the switch can be configured to process client requests, server traffic, or both. You can enable or disable processing on a port independently for each type of Layer 4 traffic.
  - Layer 4 client processing. Ports configured to process client request traffic provide address translation from the virtual IP to the real server IP address.
  - Layer 4 server processing. Ports configured to process server responses to client requests provide address translation from the real server IP address to the virtual IP address. These ports require real servers to be connected to the Web switch, directly or through a hub, router, or another switch.

NOTE – Switch ports configured for Layer 4 client/server processing can simultaneously provide Layer 2 switching and IP Routing functions.

Consider the following network topology:

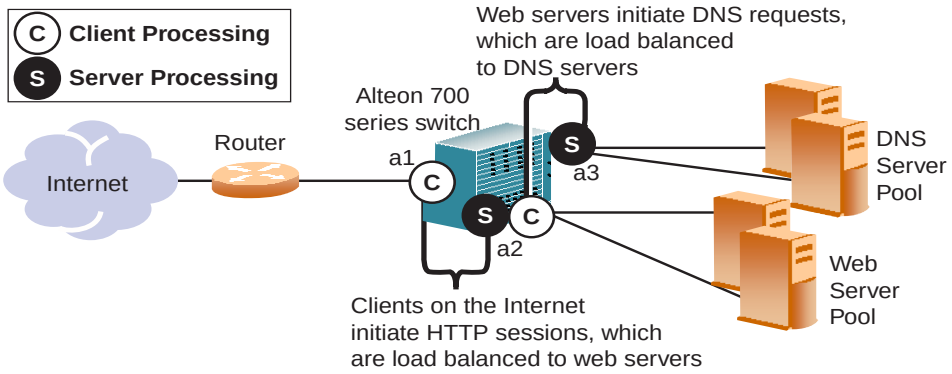


Figure 3 Example Network for Client/Server Port Configuration

In Figure 3, the switch load balances traffic to a Web server pool and to a DNS server pool. The switch port connected to the Web server pool (port a2) is asked to perform both server and client processing.

SLB Implementation

Web Hosting Configuration

Consider a situation where customer Web sites are being hosted by a popular Web hosting company and/or Internet Service Provider (ISP). The Web content is relatively static and is kept on a single NFS server for easy administration. As the customer base increases, the number of simultaneous Web connection requests also increases.

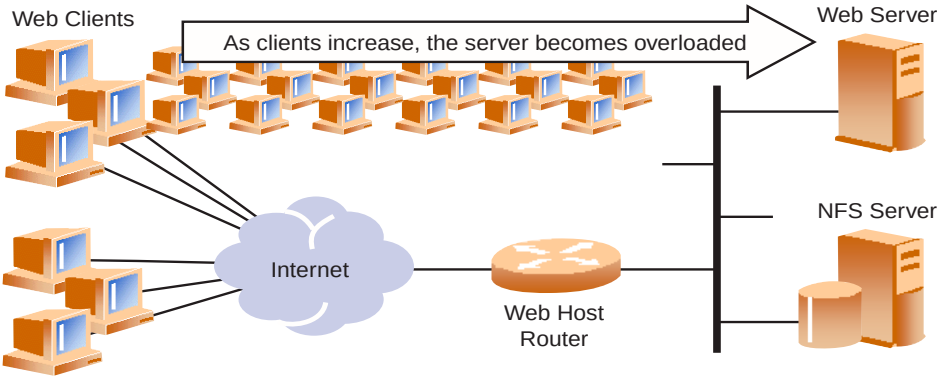


Figure 4 Web Hosting Configuration without SLB

Such a company has three primary needs:

- Increased server availability
- Server performance scalable to match new customer demands
- Easy administration of network and servers

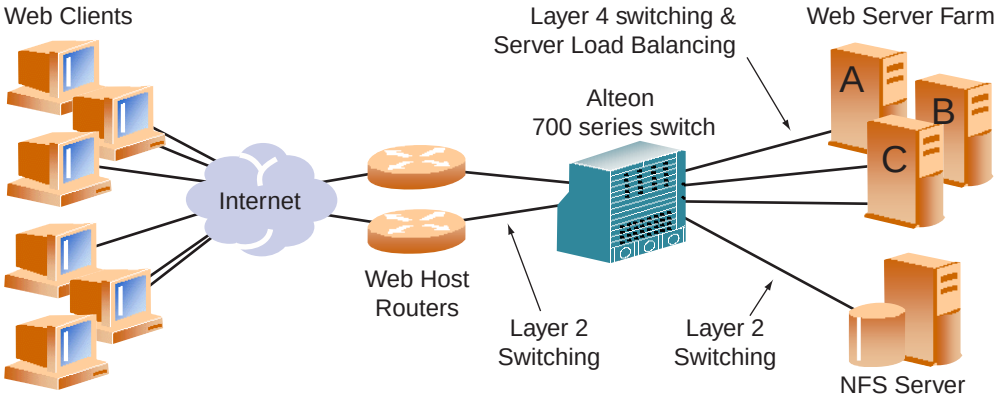


Figure 5 Web Hosting with SLB Solutions

Each concern about this company's site can be addressed by adding an Alteon WebSystems Web switch with optional SLB software.

- Reliability is increased by providing multiple paths from the clients to the Web switch, and by access to a pool of servers that have identical content. If one server fails, the others can take up the additional load.
- Performance is improved by balancing the Web request load across multiple servers. More servers can be added at any time to increase processing power.
- For ease of maintenance, servers can be added or removed dynamically without interrupting shared services.

Example Configuration for the Web Hosting Solution

In the following examples, many of the SLB options are left to their default values. See "Additional SLB Options" on page 20 for more options.

The following is required prior to configuration:

- You must be connected to the switch command-line interface as the administrator (see your *Web OS 7.0 Command Reference*).
- The optional SLB software must be enabled (see your *Web OS 7.0 Command Reference*).

NOTE – For details about any of the menu commands described in this example, see your *Web OS 7.0 Command Reference*.

---

1. **Assign an IP address to each of the real servers in the server pool.**

   The real servers in any given real server group must have an IP route to the switch that will perform the SLB functions. This is most easily accomplished by placing the switches and servers on the same IP subnet, although advanced routing techniques can be used as long as they do not violate the topology rules outlined in "Network Topology Considerations" on page 12.

   For this example, the three Web-host real servers have been given the following IP addresses on the same IP subnet:

   Table 1-1 Web Host Example: Real Server IP addresses

Real Server	IP address
Server A	200.200.200.2
Server B	200.200.200.3
Server C	200.200.200.4

NOTE – An imask option can be used to define a range of IP addresses for real and virtual servers (see "IP Address Ranges Using imask" on page 20).

---

2. **Define an IP interface on the switch.**

   The switch must have an IP route to all of the real servers which receive Web switching services. For SLB, the switch uses this path to determine the level of TCP/IP reach of the real servers.

   To configure an IP interface for this example, enter these commands from the CLI:

```
>> # /cfg/ip/if 1                          (Select IP interface #1)
>> IP Interface 1# addr 200.200.200.100    (Assign IP address for the interface)
>> IP Interface 1# enable                  (Enable IP interface #1)
```

3. **On the switch, define each real server.**

   For each real server, you must assign a real server number, specify its actual IP address, and enable the real server. For example:

```
>> IP Interface 1# /cfg/slb/real 1         (Server A is real server 1)
>> Real server 1 # rip 200.200.200.2       (Assign Server A IP address)
>> Real server 1 # enable                  (Enable real server 1)
>> Real server 1 # ../real 2               (Server B is real server 2)
>> Real server 2 # rip 200.200.200.3       (Assign Server B IP address)
>> Real server 2 # enable                  (Enable real server 2)
>> Real server 2 # ../real 3               (Server C is real server 3)
>> Real server 3 # rip 200.200.200.4       (Assign Server C IP address)
>> Real server 3 # enable                  (Enable real server 3)
```

4. **On the switch, define a real server group.**

   This combines the three real servers into one service group:

```
>> Real server 3 # /cfg/slb/group 1        (Select real server group 1)
>> Real server group 1# add 1              (Add real server 1 to group 1)
>> Real server group 1# add 2              (Add real server 2 to group 1)
>> Real server group 1# add 3              (Add real server 3 to group 1)
```

5. **On the switch, define a virtual server.**

   All client requests will be addressed to a virtual IP address on a virtual server defined on the switch. Clients acquire the virtual IP address through normal DNS resolution. In this example, HTTP is configured as the only service running on this virtual server, and this service is associated with our real server group. For example:

```
>> Real server group 1 # /cfg/slb/virt 1      (Select virtual server 1)
>> Virtual server 1# vip 200.200.200.1       (Assign a virtual server IP address)
>> Virtual server 1# enable                  (Enable the virtual server)
>> Virtual server 1# service http            (Select the HTTP service menu)
>> Virtual server 1 http Service# group 1    (Associate virtual port to real group)
```

NOTE – This configuration is not limited to HTTP Web service. Other TCP/IP services can be configured in a similar fashion. For a list of other well-known services and ports, see the command option information in your *Web OS 7.0 Command Reference*.

6. **On the switch, define the port settings.**

   In this example, the following ports are being used on the Web switch:

Table 1-2 Web Host Example: Port Usage

Port	Host	L4 Processing
a1	Server A, which serves SLB requests.	Server
a2	Server B, which serves SLB requests.	Server
a3	Server C, which serves SLB requests.	Server
a4	Back-end NFS server. All three real servers get their Web content from this machine. This port does not require Web switching features.	None
a5	Client router A. This connects the switch to the Internet where client requests originate.	Client
a6	Client router B. This also connects the switch to the Internet where client requests originate.	Client

The ports are configured as follows:

```
>> Virtual server 1# /cfg/slb/port a1        (Select physical switch port a1)
>> SLB port a1# server enable                (Enable server processing on port a1)
>> SLB port a1# ../port a2                   (Select physical switch port a2)
>> SLB port a2# server enable                (Enable server processing on port a2)
>> SLB port a2# ../port a3                   (Select physical switch port a3)
>> SLB port a3# server enable                (Enable server processing on port a3)
>> SLB port a3# ../port a5                   (Select physical switch port a5)
>> SLB port a5# client enable                (Enable client processing on port a5)
>> SLB port a5# ../port a6                   (Select physical switch port a6)
>> SLB port a6# client enable                (Enable client processing on port a6)
```

7. **On the switch, enable, apply, and verify the configuration.**

```
>> SLB port a6# ..                           (Select the SLB Menu)
>> Server Load Balancing# on                 (Turn Server Load Balancing on)
>> Server Load Balancing# apply              (Make your changes active)
>> Server Load Balancing# current            (View current settings)
```

Examine the resulting information. If any settings are incorrect, make any appropriate changes.

8. **On the switch, save your new configuration changes.**

```
>> Server Load Balancing# save               (Save for restore after reboot)
```

9. **On the switch, check the Server Load Balancing information.**

```
>> Server Load Balancing# /info/slb/dump     (View SLB information)
```

Check that all Server Load Balancing parameters are working according to expectation. If necessary, make any appropriate configuration changes and then check the information again.

Additional SLB Options

In the examples above, many of the SLB options are left to their default values. The following configuration options can be used to tune the system.

NOTE – You must `apply` any changes in order for them to take effect, and you must `save` them if you wish them remain in effect after switch reboot.

IP Address Ranges Using imask

The `imask` option lets you define a range of IP addresses for the real and virtual servers configured under SLB. By default, the `imask` setting is 255.255.255.255, which means that each real and virtual server represents a single IP address. An imask setting of 255.255.255.0 would mean that each real and virtual server represents 256 IP addresses. Consider the following example:

- A virtual server is configured with an IP address of 172.16.10.1.
- Real servers 172.16.20.1 and 172.16.30.1 are assigned to service the virtual server.
- The imask is set to 255.255.255.0.

If the client request was sent to virtual IP address 172.16.10.45, the unmasked portion of the virtual IP address (0.0.0.45) gets mapped directly to whichever real IP address is selected by the SLB algorithm. Thus, the request would be sent to either 172.16.20.45 or 172.16.30.45.

Health Checks for Real Servers

Determining health for each real server is a necessary function for SLB. By default for TCP services, the switch checks health by opening a TCP connection to each service port configured as part of each virtual service. For UDP services, the switch pings servers to determine their status.

By default, the switch checks the status of each service on each real server every two seconds. Sometimes, the real server may be too busy processing connections to respond to health checks. By default, if a service does not respond to four consecutive health checks, the switch declares the service unavailable. Both the health check interval and the number of retries can be changed:

```
>> # /cfg/slb/real <real server number>     (Select the real server)
>> Real server# inter 4                     (Check real server every 4 seconds)
>> Real server# retry 6                     (If 6 consecutive health checks fail,
                                             declare real server down)
```

Metrics for Real Server Groups

Metrics are used for selecting which real server in a group will receive the next client connection. The available metrics are `leastconns` (least connections), `minmisses` (minimum misses), `hash`, and `roundrobin`, explained in detail below. The default metric is `leastconns`.

To change a real server group metric, to `minmisses` for example, enter:

```
>> # /cfg/slb/group <group number>          (Select the real server group)
>> Real server group# metric minmisses      (Use minmisses metric)
```

Least Connections

With the `leastconns` metric, the number of connections currently open on each real server is measured in real time. The server with the fewest current connections is considered to be the best choice for the next client connection request.

This option is the most self-regulating, with the fastest servers typically getting the most connections over time, due to their ability to accept, process, and shut down connections faster than slower servers.

Minimum Misses

The `minmisses` (minimum misses) metric is optimized for Application Redirection. It uses IP address information in the client request to select a server. The specific IP address information used depends on the application:

- For Application Redirection, the client destination IP address is used. All requests for a specific IP destination address will be sent to the same server. This is particularly useful in caching applications, helping to maximize successful cache hits. Best statistical load balancing is achieved when the IP address destinations of load balanced frames are spread across a broad range of IP subnets.
- For SLB, the client source IP address and real server address are used. All requests from a specific client will be sent to the same server. This is useful for applications where client information must be retained on the server between sessions. Server load with this metric becomes most evenly balanced as the number of active clients with different source or destination addresses increases.

When selecting a server, the switch will calculate a score for each available real server based on the relevant IP address information. The server that scores the highest is assigned the connection. This metric attempts to minimize the disruption of persistency when servers are removed from service. This metric should be used only when persistence is a must.

NOTE – The minmisses metric cannot be used for Firewall Load Balancing (FWLB) since the real server IP addresses used in calculating the score for this metric are different on each side of the firewall.

Hash

The `hash` metric uses IP address information in the client request to select a server. The specific IP address information used depends on the application:

- For Application Redirection, the client destination IP address is used. All requests for a specific IP destination address will be sent to the same server. This is particularly useful for maximizing successful cache hits.
- For SLB, the client source IP address is used. All requests from a specific client will be sent to the same server. This is useful for applications where client information must be retained between sessions.
- For FWLB, both the source and destination IP address are used. This helps ensure that the two unidirectional flows of a given session to be redirected to the same firewall.

When selecting a server, a mathematical “hash” of the relevant IP address information is used as an index into the list of currently available servers. Any given IP address information will always have the same hash result, providing natural persistence as long as the server list is stable. However, if a server is added to or leaves the mix, then a different server might be assigned to a subsequent session with the same IP address information even though the original server is still available. Open connections are not cleared.

The hash metric provides more distributed load balancing than `minmisses` at any given instant. It should be used if the statistical load balancing achieved using `minmisses` is not as optimal as desired. If the load balancing statistics with `minmisses` indicate that one server is processing significantly more requests over time than other servers, consider using the `hash` metric.

Round Robin

With the `roundrobin` metric, new connections are issued to each server in turn: the first real server in this group gets the first connection, the second real server gets the next connection, followed by the third real server, and so on. When all the real servers in this group have received at least one connection, the issuing process starts over with the first real server.

Weights for Real Servers

Weights can be assigned to each real server. These weights bias load balancing to give the fastest real servers a bigger share of connections during load balancing. Weight is specified as a number from 1 to 48. Each increment increases the number of connections the real server gets. By default, each real server is given a weight setting of 1. A setting of 10 would assign the server roughly 10 times the number of connections as a server with a weight of 1. To set weights, enter the following commands:

```
>> # /cfg/slb/real <real server number>        (Select the real server)
>> Real server# weight 10                      (10 times the number of connections)
```

NOTE – Weights are not applied when using the `hash` or `minmisses` metrics.

Connection Time-outs for Real Servers

In some cases, open TCP/IP sessions might not be closed properly (for example, the switch receives the `SYN` for the session, but no `FIN` is sent). If a session is inactive for 10 minutes (the default), it is released from the switch. To change the time-out period, enter the following:

```
>> # /cfg/slb/adv                             (Select the advanced slb menu)
>> Layer 4 Advanced# timeout 4                (Specify an even numbered interval)
```

The example above would change the time-out period of all connections to 4 minutes.

Maximum Connections for Real Servers

You can set the number of open connections each real server is allowed to handle for Server Load Balancing. To set the connection limit, enter the following:

```
>> # /cfg/slb/real <real server number>       (Select the real server)
>> Real server# maxcon 1600                   (Allow 1600 connections maximum)
```

Values average between about 500 HTTP connections for slower servers, up to 1,500 for quicker, multi-processor servers. The appropriate value also depends on the duration of each session, as well as how much CPU capacity is occupied by processing each session. Connections that use a lot of Java or `CGI` scripts for forms or searches require more server resources and thus a lower `maxcon` limit. You may wish to use a performance bench-mark tool to determine how many connections your real servers can handle.

When a server reaches its `maxcon` limit, the switch no longer sends new connections to the server. When the server drops back below the `maxcon` limit, new sessions are again allowed.

Backup/Overflow Servers

A real server can backup other real servers, and can handle overflow traffic when the maximum connection limit is reached. Each backup real server must be assigned a real server number and real IP address. It must then be enabled. Finally, the backup must be assigned to each real server it will backup. The following defines real server #4 as a backup for real servers #1 and #2:

```
>> # /cfg/slb/real 4                      (Select real server #4 as backup)
>> Real server 4 # rip 200.200.200.5      (Assign backup IP address)
>> Real server 4 # enable                 (Enable real server #4)
>> Real server 4 # ../real 1              (Select real server #1)
>> Real server 1 # backup 4               (Real server #4 is backup for #1)
>> Real server 1 # ../real 2              (Select real server #2)
>> Real server 2 # backup 4               (Real server #4 is backup for #2)
```

In a similar fashion, a backup/overflow server can be assigned to a real server group. If all real servers in a real server group fail or overflow, the backup comes online.

```
>> # /cfg/slb/group <real server group number>   (Select real server group)
>> Real server group# backup r4                 (Assign real server #4 as backup)
```

Real server groups can also use another real server group for backup/overflow:

```
>> # /cfg/slb/group <real server group number>   (Select real server group)
>> Real server group# backup g2                 (Assign group #2 as backup)
```

Mapping Virtual Ports to Real Ports

In addition to providing direct real server access in some situations (see “Port Mapping” on page 29), mapping is required when administrators choose to execute their real server processes on different TCP/UDP ports than the well known TCP/UDP ports. Otherwise, virtual server ports are mapped directly to real server ports by default and require no mapping configuration.

Port mapping is configured from the virtual server services menu. For example, to map the virtual server TCP/UDP port 80 to real server TCP/UDP port 8004, you could enter the following:

```
>> # /cfg/slb/virt 1/service 80              (Select virtual server port 80)
>> Virtual Server 1 http Service# rport 8004 (map to real port 8004)
```

NOTE – Port mapping is supported with Direct Access Mode when filtering is enabled, a proxy IP address is configured, or URL parsing is enabled on any switch port. For information about DAM, refer to “Direct Access Mode” on page 28.

Proxy IP Addresses for Complex SLB Networks

For standard SLB, all client-to-server requests to a particular virtual server and all related server-to-client responses must pass through the same Web switch.

In complex network topologies, routers and other devices can create alternate paths around the Web switch managing SLB functions (see Figure 2 on page 12). Under such conditions, the client switch ports can use a proxy IP address.

When the client requests services from the switch's virtual server, the client sends its own IP address for use as a return address. If a proxy IP address is configured for the client port on the switch, the switch replaces the client's source IP address with the switch's own proxy IP address before sending the request to the real server. This creates the illusion that the switch originated the request. The real server uses the switch's proxy IP address as the destination address for any response. This forces the SLB traffic to return through the proper switch, regardless of alternate paths. Once the switch receives the proxied data, it puts the original client IP address into the destination address and sends the packet to the client. This process is transparent to the client.

NOTE – Because requests appear to come from the switch proxy IP address rather than the client source IP address, use of proxy addresses can generate misleading information for network statistics or debugging.

The proxy IP address can also be used for direct access to the real servers (see "Direct Client Access to Real Servers" on page 28).

The following procedure can be used for configuring proxy IP addresses.

1. **Disable server processing on affected switch ports.**

   When implementing proxies, switch ports can be re-configured to disable server processing. Re-examining the "Example Configuration for the Web Hosting Solution" on page 16, the following revised port conditions are used:

Table 1-3 Proxy Example: Alteon 708 Port Usage

Port	Host	L4 Processing
a1	Server A	None
a2	Server B	None
a3	Server C	None
a4	Back-end NFS server. All three real servers get their Web content from this machine. This port does not require Web switching.	None
a5	Client router A. This connects the switch to the Internet, where all client requests originate.	Client
a6	Client router B. This also connects the switch to the Internet, where all client requests originate.	Client

The following commands are used to disable server processing on ports 1-3:

```
>> # /cfg/slb/port a1                 (Select switch port a1)
>> SLB port a1# server dis           (Disable server processing on port a1)
>> SLB port a1# ../port a2           (Select switch port a2)
>> SLB port a2# server dis           (Disable server processing on port a2)
>> SLB port a2# ../port a3           (Select switch port a3)
>> SLB port a3# server dis           (Disable server processing on port a3)
```

2. **Add proxy IP addresses to the client ports.**

   Each "`client`" ports requires a proxy IP address. Each proxy IP address must be unique on your network. The following shows commands used to configure client proxies for this example:

```
>> # /cfg/slb/port a5                    (Select network port a5)
>> SLB port a5# pip 200.200.200.68      (Set proxy IP address for client port a5)
>> SLB port a5# ../port a6              (Select network port a6)
>> SLB port a6# pip 200.200.200.69      (Set proxy IP address for client port a6)
```

   The proxies are transparent to the user.

3. **Apply and save your changes.**

NOTE – Remember that you must `apply` any changes in order for them to take effect, and you must `save` them if you wish them remain in effect after switch reboot. Also, the `/info/slb` command is useful for checking the state of Server Load Balancing operations.

Direct Client Access to Real Servers

Some clients may need direct access to the real servers, to, for example, monitor a real server from a management workstation. This access can be provided in a number of ways:

- Direct Access Mode
- Multiple IP addresses on the server
- Proxy IP addresses
- Port mapping
- Management network

Direct Access Mode

When Direct Access Mode (`/cfg/slb/direct`) is enabled on a switch, any client can communicate with any real server's load-balanced service. Also, in Direct Access Mode, any number of virtual services can be configured to load balance a real service.

Traffic sent directly to real server IP addresses is excluded from load balancing decisions. The same clients may also communicate to the virtual server IP address for load balanced requests.

NOTE – When Direct Access Mode is enabled on a switch, port mapping and default gateway load balancing is supported only when filtering is enabled, a proxy IP address is configured, or URL parsing is enabled on any switch port.

Multiple IP Addresses on the Server

One way to provide both SLB access and direct access to a real server is to assign multiple IP addresses to the real server. For example, one IP address could be established exclusively for SLB, and another could be used for direct access needs.

Proxy IP Addresses

Proxy IP addresses are used primarily to eliminate SLB topology restrictions in complex networks (see "Network Topology Considerations" on page 12). Proxy IP addresses can also provide direct access to real servers.

If the switch port to the client is configured with a proxy IP address (see "Proxy IP Addresses for Complex SLB Networks" on page 25), the client can access each real server directly using the real server's IP address. This requires that the switch port connected to the real server has server and client processing disabled (see the `server` and `client` options under `/cfg/slb/port` in your *Web OS 7.0 Command Reference*).

SLB is still accessed using the virtual server IP address.

Port Mapping

When SLB is used without proxy IP addresses and without Direct Access Mode (DAM), the switch must process the server-to-client responses. If a client were to access the real server IP address and port directly, bypassing client processing, the server-to-client response could be mishandled by SLB processing as it returns through the switch, with the real server IP address getting re-mapped back to the virtual server IP address.

First, two port processes must be executed on the real server. One real server port will handle the direct traffic, and the other will handle SLB traffic. Then, the virtual server port must be mapped to the proper real server port.

In the following figure, clients can access SLB services through well-known TCP port 80 at the virtual server's IP address. This is mapped to TCP port 8000 on the real server. For direct access that bypasses the virtual server and SLB, clients can specify well-known TCP port 80 at the real server's IP address.

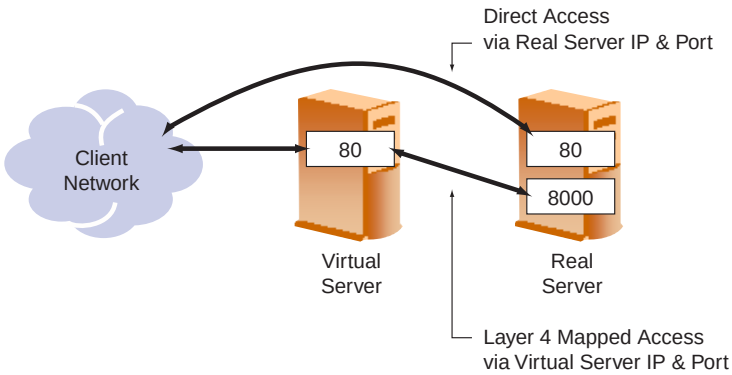


Figure 6 Mapped and Non-mapped server access

NOTE – Port mapping is supported with Direct Access Mode when filtering is enabled, a proxy IP address is configured, or URL parsing is enabled on any switch port.

Management Network

Typically, the management network is used by network administrators to monitor real servers and services. By configuring the `mnet` and `mmask` options of the SLB Configuration Menu (`cfg/slb`) you can access the real services being load balanced.

NOTE – Clients on the management network do not have access to SLB services and cannot access the virtual services being load balanced.

The `mnet` and `mmask` options are described below:

- `mnet`: If defined, management traffic with this source IP address will be allowed direct (non-SLB) access to the real servers. Only specify an IP address in dotted decimal notation. A range of IP addresses is produced when used with the `mmask` option
- `mmask`: This IP address mask is used with `mnet` to select management traffic which is allowed direct real server access only.

CHAPTER 2

Application Redirection

Application Redirection improves network bandwidth and provides unique network solutions. Filters can be created to redirect traffic to cache and application servers. Repeated client access to common web or application content across the Internet can be an inefficient use of network resources. By redirecting client requests to a local web-cache or application server, you increase the speed at which clients access the information and free up valuable network bandwidth.

NOTE – To access Application Redirection functionality, the optional Layer 4 software must be enabled in the switch (see "Filtering and Layer 4" in Chapter 7 of the *Web OS 7.0 Command Reference*).

Web-Cache Redirection Example

Much of the information downloaded from the Internet is not unique, as clients will often access the same information many times as they return to a Web page for additional information or to explore other links. Duplicate information also gets requested as the components that make up Internet data at a particular website (pictures, buttons, frames, text, and so on) are reloaded from page to page. When you consider this scenario in the context of many clients, it becomes apparent that redundant requests can consume a considerable amount of your available bandwidth to the Internet.

Web-cache redirection can help alleviate the congestion seen at your Internet router. When Application Redirection filters are properly configured for your Web OS-powered switch, outbound client requests for Internet data are intercepted and redirected to a group of web-cache servers on your network. The web-cache servers duplicate and store inbound Internet data that has been requested by your clients. If the web-cache servers recognize a client's outbound request as one that can be filled with cached information, the web-cache servers will supply the information, rather than sending the request out across the Internet.

In addition to increasing the efficiency of your network, access to locally cached information can be granted much faster than by requesting the same information across the Internet.

Web-Cache Redirection Environment

Consider a network where client HTTP requests begin to regularly overload the Internet router.

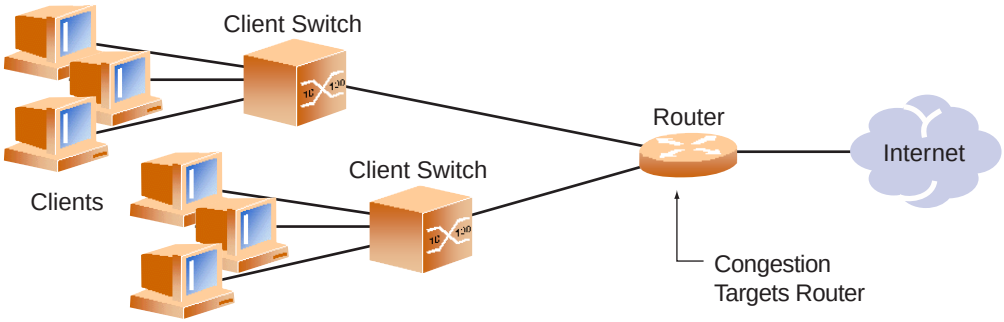


Figure 7 Traditional network without Web Cache Redirection

The network needs a solution that addresses the following key concerns:

- The solution must be readily scalable.
- The administrator should not have to reconfigure all the clients' browsers to use Proxy Servers.

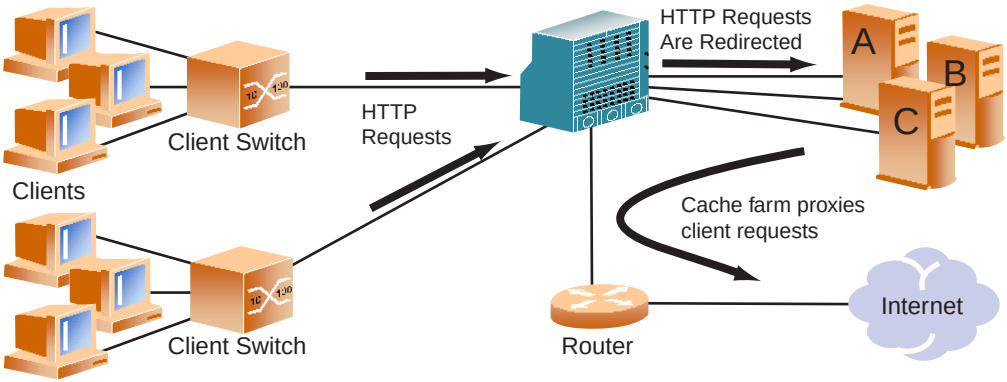


Figure 8 Network with Web Cache Redirection

Adding an Alteon WebSystems switch with optional Layer 4 software addresses these issues:

- Web-cache servers can be added or removed dynamically without interrupting services.
- Performance is improved by balancing the cached web request load across multiple servers. More servers can be added at any time to increase processing power.
- The proxy is transparent to the client.
- Frames that are not associated with HTTP requests are passed normally to the router.

Example Configuration for the Web-Cache Solution

The following is required prior to configuration:

- You must be connected to the switch command-line interface as the administrator.
- Optional Layer 4 software must be enabled.

NOTE – For details about the procedures above, and about any of the menu commands described in this example, see the *Web OS 7.0 Command Reference*.

In this example, an Alteon 700 series switch is placed between the clients and the border gateway to the Internet. The switch will be configured to intercept all Internet bound HTTP requests (on default TCP port 80), and redirect them to the web-cache servers. The switch will distribute HTTP requests equally to the web-cache servers based on the destination IP address of the requests.

Also, filters are not limited to the few protocols and TCP or UDP applications shown in this example. See the *Web OS 7.0 Command Reference* for a list of other well-known protocols and services.

1. **Assign an IP address to each of the web-cache servers.**

   Just as with Server Load Balancing, the web-cache real servers will be assigned an IP address and placed into a real server group. The real servers must be in the same VLAN and must have an IP route to the switch that will perform the web-cache redirection. In addition, the path from the switch to the real servers must not contain a router. The router would stop HTTP requests from reaching the web-cache servers, instead directing them back out to the Internet.

   More complex network topologies can be used if configuring IP proxy addresses (see "IP Proxy Addresses for Transparent Proxies or Complex Networks" on page 38).

   For this example, the three web-cache real servers have the following IP addresses on the same IP subnet:

   Table 2-1 Web-Cache Example: Real Server IP addresses

Web Cache Server	IP address
Server A	200.200.200.2
Server B	200.200.200.3
Server C	200.200.200.4

2. **Install or configure web-cache servers in transparent mode.**

   Or, define proxy IP addresses on the switch (see "IP Proxy Addresses for Transparent Proxies or Complex Networks" on page 38).

   Full Network Address Translation (NAT) is required.

3. **Define an IP interface on the switch.**

   Because, by default, the switch only re-maps destination MAC addresses, it must have an IP interface on the same subnet as the three web-cache servers.

   To configure an IP interface for this example, enter this command from the CLI:

```
>> Main# /cfg/ip/if 1                        (Select IP interface #1)
>> IP Interface 1# addr 200.200.200.100      (Assign IP address for the interface)
>> IP Interface 1# enable                    (Enable IP interface #1)
```

NOTE – The IP interface and the real servers must be in the same subnet. This example assumes that all ports and IP interfaces use default VLAN #1, requiring no special VLAN configuration for the ports or IP interface.

4. **On the switch, define each Real Server.**

   For each web-cache real server, you must assign a real server number, specify its actual IP address, and enabled the real server. For example:

```
>> ip# /cfg/slb/real 1                       (Server A is real server 1)
>> Real server 1 # rip 200.200.200.2         (Assign Server A IP address)
>> Real server 1 # enable                    (Enable real server 1)
>> Real server 1 # ../real 2                 (Server B is real server 2)
>> Real server 2 # rip 200.200.200.3         (Assign Server B IP address)
>> Real server 2 # enable                    (Enable real server 2)
>> Real server 2 # ../real 3                 (Server C is real server 3)
>> Real server 3 # rip 200.200.200.4         (Assign Server C IP address)
>> Real server 3 # enable                    (Enable real server 3)
```

5. **On the switch, define a Real Server Group.**

   This places the three web-cache real servers into one service group:

```
>> Real server 3 # /cfg/slb/group 1          (Select real server group 1)
>> Real server group 1 # add 1               (Add real server 1 to group 1)
>> Real server group 1 # add 2               (Add real server 2 to group 1)
>> Real server group 1 # add 3               (Add real server 3 to group 1)
```

6. **On the switch, set the Real Server Group metric to `minmisses`.**

   This helps minimize web-cache misses in the event real servers fail or are taken out of service:

```
>> Real server group 1 # metric minmisses  (Metric for minimum cache misses.)
```

7. **On the switch, verify that server processing is disabled on the ports supporting application redirection.**

NOTE – Do not use the "server" setting on a port with Application Redirection enabled. Server processing is used only with Server Load Balancing. To disable server processing on the port, use the commands on the **`/cfg/slb/port`** menu, as described in Chapter 8 of the *Web OS 7.0 Command Reference*.

---

8. **On the switch, create a filter that will intercept and redirect all client HTTP requests.**

   The filter must be able to intercept all TCP traffic for the HTTP destination port, and must redirect it to the proper port on the real server group:

```
>> SLB port a6 # /cfg/slb/filt 2          (Select the menu for Filter #2)
>> Filter 2# sip any                      (From any source IP addresses)
>> Filter 2# dip any                      (To any destination IP addresses)
>> Filter 2# proto tcp                    (For TCP protocol traffic)
>> Filter 2# sport any                    (From any source port)
>> Filter 2# dport http                   (To an HTTP destination port)
>> Filter 2# action redir                 (Set the action for redirection)
>> Filter 2# rport http                   (Set the redirection port)
>> Filter 2# group 1                      (Select real server group 1)
>> Filter 2# enable                       (Enable the filter)
```

   The `rport` parameter must be configured whenever TCP/UDP protocol traffic is redirected. The `rport` parameter defines the real server TCP or UDP port to which redirected traffic will be sent. The port defined by the `rport` parameter is used when performing Layer 4 health checks of TCP services.

Also, if transparent proxies are used for Network Address Translation (NAT) on the switch (see Step on page 34), the `rport` parameter must be configured for all Application Redirection filters. Take care to use the proper port designation with `rport`: if the transparent proxy operation resides on the host, the well-known port (80, or "http") is probably required. If the transparent proxy occurs on the switch, make sure to use the service port required by the specific software package.

See "IP Proxy Addresses for Transparent Proxies or Complex Networks" on page 38 for more about IP proxy addresses.

9. **On the switch, create a default filter.**

In this case, the default filter will allow all non-cached traffic to proceed normally:

```
>> Filter 2# ../filt 224                    (Select the default filter)
>> Filter 224# sip any                      (From any source IP addresses)
>> Filter 224# dip any                      (To any destination IP addresses)
>> Filter 224# proto any                    (For any protocols)
>> Filter 224# action allow                 (Set the action to allow traffic)
>> Filter 224# enable                       (Enable the default filter)
```

NOTE – When the `proto` parameter is not tcp or udp, then `sport` and `dport` are ignored.

10. **On the switch, assign the filters to the client ports.**

Assuming that the redirected clients are connected to physical switch ports 5 and 6, both ports are configured to use the previously created filters as follows:

```
>> Filter 224# ../port a5                   (Select the SLB port a5)
>> SLB port a5 # add 2                      (Add filter 1 to port a5)
>> SLB port a5 # add 224                    (Add the default filter to port a5)
>> SLB port a5 # filt enable                (Enable filtering for port a5)
>> SLB port a5 # ../port a6                 (Select the SLB port a6)
>> SLB port a6 # add 2                      (Add filter 1 to port a6)
>> SLB port a6 # add 224                    (Add the default filter to port a6)
>> SLB port a6 # filt enable                (Enable filtering for port a6)
```

11. **On the switch, enabled, apply, and verify the configuration.**

```
>> SLB port a6 # ..                          (Select Server Load Balancing Menu)
>> Server Load Balancing# on                 (Activate Layer 4 software services)
>> Server Load Balancing# apply              (Make your changes active)
>> Server Load Balancing# cur                (View current settings)
```

NOTE – Server Load Balancing must be turned on in order for Application Redirection to work properly. The "`on`" command is valid only if the optional Layer 4 software is enabled on your switch (see "Activating Optional Software" in the *Web OS 7.0 Command Reference*.

12. **Examine the resulting information from the "`current`" command. If any settings are incorrect, make appropriate changes.**

13. **On the switch, save your new configuration changes.**

```
>> Server Load Balancing# save               (Save for restore after reboot)
```

14. **On the switch, check the Server Load Balancing information.**

```
>> Server Load Balancing# /info/slb/dump     (View all SLB information)
```

Check that all Server Load Balancing parameters are working according to expectation. If necessary, make any appropriate configuration changes and then check the information again.

NOTE – Changes to filters on a given port only effect new sessions. To make filter changes take effect immediately, clear the session binding table for the port (see the `/oper/slb/clear` command in the *Web OS 7.0 Command Reference*).

IP Proxy Addresses for Transparent Proxies or Complex Networks

Transparent proxies provide the benefits listed below when used with Application Redirection. Application redirection is automatically enabled when a filter with the `redir` action is applied on a port.

- With proxies IP addresses configured on redirected ports, the switch can redirect client requests to servers located on any subnet, anywhere.
- The switch can perform transparent substitution for all source and destination addresses, including destination port remapping. This provides support for comprehensive, fully-transparent proxies. These proxies are transparent to the user. No additional client configuration is needed.

The following procedure can be used for configuring proxy IP addresses:

1. **Add proxy IP addresses to the redirection ports.**

   Each of the ports using redirection filters require proxy IP addresses to be configured. Each proxy IP address must be unique on your network. These are configured as follows:

```
>> SLB port a3# /cfg/slb/port a5          (Select network port a5)
>> SLB port a5# pip 200.200.200.68        (Set proxy IP address for port a5)
>> SLB port a5# ../proxy enable           (Enable proxy port a5)
>> SLB port a5# ../port a6                (Select network port a6)
>> SLB port a6# pip 200.200.200.69        (Set proxy IP address for port a6)
>> SLB port a6# ../proxy enable           (Enable proxy port a6)
```

2. **Configure the Application Redirection filters.**

   Once proxy IP addresses are established, you need to configure each Application Redirection filter (filter 2 in our example) with the real server TCP or UDP port to which redirected traffic will be sent. In this case, we are mapping the requests to different destination port (8080). You must also enable proxies on the real servers:

```
>> # /cfg/slb/filt 2                          (Select the menu for Filter #2)
>> Filter 2 # rport 8080                      (Set proxy redirection port)
>> Filter 2 # real 1/proxy enable             (Enable proxy on real servers)
>> Real server 1 # ../real 2/proxy enable     (Enable proxy on real servers)
>> Real server 2 # ../real 3/proxy enable     (Enable proxy on real servers)
```

NOTE – This configuration is not limited to HTTP Web service. Other TCP/IP services can be configured in a similar fashion. For example, if this had been a DNS redirect, `rport` would be sent to well-known port 53 (or the service port you want to remap to). For a list of other well-known services and ports, see the *Web OS 7.0 Command Reference*.

3. **Apply and save your changes.**

4. **Check server statistics to verify that traffic has been redirected based on filtering criteria:**

```
>> # /info/slb/group <group#>/filter <filter#>(View statistics for server filter)
```

Excluding Non-Cacheable Sites

Some Web sites provide content which isn't well suited for redirection to cache servers. Such sites might provide browser-based games, applications that keep real-time session information or authenticate by client IP address.

To prevent such sites from being redirected to cache-servers, create a filter which allows this specific traffic to pass normally through the switch. This filter must have a higher precedence (a lower filter number) than the Application Redirection filter.

For example, if you wished to prevent a popular web-based game site on subnet 200.10.10.* from being redirected, you could add the following to the previous example configuration:

```
>> # /cfg/slb/filt 1                    (Select the menu for Filter #1)
>> Filter 1# dip 200.10.10.0             (To the site's destination IP address)
>> Filter 1# dmask 255.255.255.0         (For entire subnet range)
>> Filter 1# sip any                     (From any source IP address)
>> Filter 1# proto tcp                   (For TCP traffic)
>> Filter 1# dport http                  (To an HTTP destination port)
>> Filter 1# sport any                   (From any source port)
>> Filter 1# action allow                (Allow matching traffic to pass)
>> Filter 1# enable                      (Enable the filter)
>> Filter 1# ../port a5                  (Select SLB port a5)
>> SLB port a5# add 1                    (Add the filter to port a5)
>> SLB port a5# ../port a6               (Select SLB port a6)
>> SLB port a6# add 1                    (Add the filter to port a6)
>> SLB port a6# apply                    (Apply configuration changes)
>> SLB port a6# save                     (Save configuration changes)
```

Defining IP Address Ranges for Filters

You can specify a range of IP address for filtering both the source and/or destination IP address for traffic. When a range of IP addresses is needed, the `sip` (source) or `dip` (destination) defines the base IP address in the desired range, and the `smask` (source) or `dmask` (destination) is the mask which is applied to produce the range.

For example, to determine if a client request's destination IP address should be redirected to the cache servers attached to a particular switch, the destination IP address is masked (bit-wise AND) with the `dmask` and then compared to the `dip`.

As another example, you could configure the switch with two filters so that each would handle traffic filtering for one half of the Internet. To do this, you could define the following parameters:

Table 2-2 Filtering IP Address Ranges

Filter	Internet Address Range	dip	dmask
#1	0.0.0.0 - 127.255.255.255	0.0.0.0	127.0.0.0
#2	127.0.0.0 - 255.255.255.255	127.0.0.0	127.0.0.0

Index

Symbols

[ ] 6

A

application redirection 31
client IP address authentication 39
example with NAT 34, 36
games and real-time applications 39
proxies ..... 32, 36 to 38
rport 35, 38
topologies 33
web-cache redirection example ..... 31 to 39

B

backup servers 24

C

CGI-bin scripts 12, 23
client traffic processing 13
SLB web balancing example 18
commands
conventions used in this manual 6
configuration
imask 20
contacting Alteon WebSystems 7
customer support 7

D

dip (destination IP address for filtering) 40
direct real server access 28
dmask
destination mask for filtering 40
Domain Name System (DNS)
round robin 10
dport (filtering option) 36

F

failed server protection, SLB 9
fault tolerance
Server Load Balancing 15
filtering
proto (option) 36
filters
IP address ranges 40

H

hash metric 22
health checks 35

I

Internet Service Provider (ISP), SLB configuration example 14
IP address
filter ranges 40
proxies ..... 25, 32, 36 to 38
real server groups 17
real servers 11, 16
SLB real servers 17
virtual servers 11, 12, 18
IP interface
example configuration 17
IP proxies
for application redirection 38
for Server Load Balancing 25
See also proxies, proxy IP address (PIP).
IP routing 13

L

Layer 4
optional software 9
least connections (SLB Real Server metric) 21

M

manual style conventions 6
mapping ports 38
mapping virtual ports to real ports....24
maxcons limit 24
maximum connections 23, 24
mcon (maximum connections)....23
minimum misses (SLB real server metric) 21

N

Network Address Translation (NAT) 34, 36
network performance
statistics, with use of proxy addresses....25
NFS server 14
non-cacheable sites
application redirection
non-cacheable sites39
none (port processing mode)
SLB web balancing example....18

O

optional software....31
Layer 4 SLB support....9
overflow servers....24

P

persistent bindings....12
PIP. *See* proxies, proxy IP address.
port mapping 38
port processing mode
client....18
none....18
server....13, 18
ports
mapping 29
SLB configuration example 18
proxies .... 25, 32 to 38
proxy IP address (PIP) 25, 28, 38
proxy servers 32

R

real server groups
backup/overflow servers 24
configuration example 17
real servers.... 12
backup/overflow servers 24
connection timeouts 23
maximum connections.... 23
SLB configuration example.... 17
weights.... 23
redirection. *See* application redirection
roundrobin
SLB Real Server metric 22
routers
using redirection to reduce Internet congestion . 31
web-cache redirection example.... 32
rport
filtering.... 35, 38

S

scalability, service 9
server (port processing mode)
SLB web balancing example 18
Server Load Balancing
backup servers.... 24
complex network topologies.... 25
configuration example 14, 25
direct real server access 28
failed server protection 9
fault tolerance.... 15
maximum connections.... 23
overflow servers 24
overview.... 10
persistent bindings 12
port processing modes 13, 18
proxies 25
proxy IP addresses 28
real server group 17
real server IP address (RIP).... 11
real servers.... 12
topology considerations 12
virtual IP address (VIP) 11, 12
virtual servers.... 11, 18
weights.... 23
server pool.... 9
server port processing 13
shared services 9

SIP (source IP address for filtering) 40
smask
    source mask for filtering 40
sport (filtering option) 36

T

TCP
    health checking using 20
    port 80 29
TCP/UDP
    port numbers 24
text conventions 6
timeouts
    real server connnections 23
transparent proxies ..... 25, 32, 36 to 38
typographic conventions, manual 6

U

UDP
    server status using 20

V

virtual IP address (VIP) 11, 12
virtual servers 11
    configuration example 18
    IP address 18
VLANs 33
    VLAN #1 (default) 34

W

web hosting 14
web-cache redirection. *See* application redirection
web-cache servers ..... 31 to 33, 34
weights 23