Web OS Switch Software



7.2 Application Guide

Part Number: 050156, Revision A, June 2001

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 2001 Alteon WebSystems, Inc., 50 Great Oaks Boulevard, San Jose, California 95119, USA. All rights reserved. Part Number: 050156, 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 and Alteon 708 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 11
Who Should Use This Guide 11
What You'll Find in This Guide 11
Typographic Conventions 13
Contacting Alteon WebSystems 14

Part 1: Basic Switching and Routing

Chapter 1: Basic IP Routing 17
IP Routing Benefits 18
Routing Between IP Subnets 18
Example of Subnet Routing 21
Defining IP Address Ranges for the Local Route Cache 25
DHCP Relay 26
- DHCP Overview 26
- Configuring a DHCP Relay Agent 27

Chapter 2: VLANs 29
VLAN ID Numbers 30
VLAN Tagging 30
VLANs and the IP Interfaces 31
VLAN Topologies and Design Issues 31
- Example 1: Multiple VLANS with Tagging Adapters 32
- Example 2: Parallel Links with VLANs 34

VLANs and Multiple Spanning Trees 35

Chapter 3: Port Trunking 37
Port Trunking Overview 38
Statistical Load Distribution 38
Built-In Fault Tolerance 39
Port Trunking Example 40

Chapter 4: OSPF 43
OSPF Overview 44
Types of OSPF Areas 44
Types of OSPF Routing Devices 45
Neighbors and Adjacencies 46
The Link-State Database 47
The Shortest Path First Tree 47
Authentication 47
Internal Versus External Routing 47
Web OS 7.2 OSPF Implementation 48
Configurable Parameters 48
Defining Areas 49
Interface Cost 52
Electing the Designated Router and Backup 52
Summarizing Routes 52
Default Routes 53
Virtual Links 54
Router ID 54
Authentication 55
Host Routes for Load Balancing 55
Redistributing Routes into OSPF 56
OSPF Features Not Supported in This Release 59
OSPF Configuration Examples 59
Example 1: Simple OSPF Domain 60
Example 2: Virtual Links 62
Example 3: Summarizing Routes 66
Example 4: Host Routes 68
Verifying OSPF Support 73

Chapter 5: Border Gateway Protocol 75
Internal Routing Versus External Routing 76
Forming BGP Peer Routers 77
What is a Route Map? 78
Incoming and Outgoing Route Maps 79
Precedence 79
Configuration Overview 79
Aggregating Routes 81
Redistributing Routes 82
BGP Attributes 83
Local Preference Attribute 83
Metric (Multi-Exit Discriminator) Attribute 83
Selecting Route Paths in BGP 84
Configuration Examples 85
BGP Failover Example 85
Default Redistribution and Route Aggregation Example 88
Web OS BGP Limitations 89

Chapter 6: Secure Switch Management 91
Setting Allowable Source IP Address Range 92
Secure Switch Management 93
Authentication and Authorization Protocols 93
Requirements 94
RADIUS Authentication and Authorization 94
RADIUS Authentication Features in Web OS 96
Web Switch User Accounts 97
Secure Shell and Secure Copy 99
Encryption of Management Messages 100
SCP Services 100
RSA Host and Server Keys 101
Radius Authentication 102
Secure ID Support 102

Part 2: Web Switching Fundamentals

Chapter 7: Server Load Balancing 107

Understanding Server Load Balancing 108
- Benefits 108
- Identifying Your Network Needs 108
- How Server Load Balancing Works 109

Network Topology Requirements 110

Configuring Server Load Balancing 112

Additional SLB Options 117
- IP Address Ranges Using imask 117
- Health Checks for Real Servers 118
- Metrics for Real Server Groups 118
- Weights for Real Servers 120
- Connection Time-outs for Real Servers 120
- Maximum Connections for Real Servers 121
- Backup/Overflow Servers 121
- Mapping Virtual Server Ports to Real Server Ports 122
- Direct Server Return 122
- Proxy IP Addresses for Complex SLB Networks 123
- Disabling Real Servers 125
- IP Server Load Balancing 125

Direct Client Access to Real Servers 126
- Direct Access Mode 126
- Multiple IP Addresses on the Server 126
- Proxy IP Addresses 126
- Port Mapping 127
- Management Network 128

Chapter 8: Filtering 129
Filtering Overview 130
Filtering Benefits 130
Filtering Criteria 130
Defining IP Address Ranges for Filters 132
Stacking Filters 132
Overlapping Filters 133
The Default Filter 133
Optimizing Filter Performance 134
Filter Logs 134
Security Example 135
Network Address Translation (NAT) Examples 141
Internal Client Access to the Internet 141
External Client Access to a Server 142

Chapter 9: Application Redirection 145
Overview 146
Web Cache Redirection Environment 146
Additional Application Redirection Options 147
Web Cache Configuration Example 148
IP Proxy Addresses for NAT 153
Excluding Noncacheable Sites 154

Chapter 10: Health Checking 155
Real Server Health Check Parameters 156
Health Check Types 157
TCP Health Checks 157
Application-Specific Health Checks 157
ICMP Health Checks 157
Script-Based Health Checks 157
Hostname for HTTP Content Health Checks 158
RADIUS Server Health Checks 160
IMAP Server Health Checking 161

Script-Based Health Checks 161
Script Format 162
Scripting Guidelines 163
Script Configuration Example 164
Verifying Script-Based Health Checks 166
HTTPS/SSL Health Check 166
Failure Types 167
Service Failure 167
Server Failure 167

Chapter 11: High Availability 169

VRRP Overview 170
VRRP Components 170
VRRP Operation 173
Selecting the Master VRRP Router 173
Active-Standby Failover 174
Failover Methods 175
Active-Standby Redundancy 176
Active-Active Redundancy 177
Hot-Standby Redundancy 177
Synchronizing Configurations 180
Web OS Extensions to VRRP 181
Virtual Server Routers 181
Sharing/Active-Active Failover 182
Tracking VRRP Router Priority 183
High Availability Configurations 184
Active-Standby Virtual Server Router Configuration 185
Active-Active VIR and VSR Configuration 187
Active/Active Server Load Balancing Configuration 189
VRRP-Based Hot-Standby Configuration 197

Virtual Router Deployment Considerations 199
Mixing Active-Standby and Active-Active Virtual Routers 199
Synchronizing Active/Active Failover 199
Eliminating Loops with STP and VLANs 200
Assigning VRRP Virtual Router ID 202
Configuring the Switch for Tracking 202
Synchronizing Configurations 203

Part 3: Advanced Web Switching

Chapter 12: Global Server Load Balancing 207
GSLB Overview 208
Benefits 208
How GSLB Works 209
Configuring GSLB 211
Step 1: Configure the Basics at the California Site 213
Step 2: Configure the California Switch for Standard SLB 214
Step 3: Configure the California Site for GSLB 216
Step 4: Configure the Basics at the Denver Site 217
Step 5: Configure the Denver Switch for Standard SLB 218
Step 6: Configure the Denver Site for GSLB 220
Using Proxy IP for Non-HTTP Redirects 222
How Proxy IP Works 223
Configuring IP Proxy Address 224
Verifying GSLB Operation 225
Configuring Client Site Preferences 226
Using Border Gateway Protocol for GSLB 229

Chapter 13: Firewall Load Balancing 231
Overview 232
Basic FWLB Implementation 233
Configuring Basic FWLB 235
Adding a DMZ 242
Firewall Health Checks 244
Firewall Service Monitoring 244
Physical Link Monitoring 244
Using HTTP Health Checks 245
FWLB with High Availability Checklist 246
Primary Switch 246
Secondary Switch 246
Firewall Configuration Examples 247
Four Subnet FWLB Using Alteon 708 Web Switches 247
Four Switches, Two Routing Firewalls, with No Interior Hubs 253
Configuring NAT on Solaris Firewalls 260
NAT for Single Devices on the Clean-Side Network 261
Static NAT, with NAT Performed by Firewall 264

Index 267

Preface

This *Application Guide* describes how to configure and use the Web OS 7.2 software on the Alteon 708.

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

Who Should Use This Guide

This *Application Guide* is intended for network installers and system administrators engaged in configuring and maintaining a network. The administrator should be familiar with Ethernet concepts, IP addressing, the IEEE 802.1d Spanning Tree Protocol, and SNMP configuration parameters.

What You'll Find in This Guide

The chapters in this book will help you plan, implement, and administer the use of Web OS 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 708.

Part 1: Basic Switching and Routing

- Chapter 1, "Basic IP Routing," explains Layer 2 switching and IP routing. This chapter also provides a step-by-step procedure on how to configure IP interfaces on the Web switch.
- Chapter 2, "VLANs," explains the concept of (Virtual Local Area Networks) VLANs and how to configure VLANs on the switch.
- Chapter 3, "Port Trunking," provides configuration examples for trunking multiple ports together.

- Chapter 4, “OSPF,” discusses OSPF support with configuration examples for creating a simple OSPF domain, summarizing routes, and defining virtual links.
- Chapter 5, “Border Gateway Protocol,” explains route maps, peer routers, and how BGP selects the best route to a network. This chapter also provides configuration examples for route aggregation and default route redistribution.
- Chapter 6, “Secure Switch Management,” describes secure management of the switch using source IP address ranges, RADIUS authentication, and using secure tunnels for encrypted messages.

Part 2: Web Switching Fundamentals

- Chapter 7, “Server Load Balancing,” discusses the benefits of server load balancing and how it works. This chapter provides a step-by-step procedure on how to configure your switch for server load balancing.
- Chapter 8, “Filtering,” provides an example of configuring filters for security. This chapter also describes the TCP ACK filter criteria that provides greater flexibility and Network Address Translation (NAT) examples.
- Chapter 9, “Application Redirection,” describes how redirecting client requests to a local Web cache or application server can increase the speed at which clients access the information and free up valuable network bandwidth.
- Chapter 10, “Health Checking,” explains how to customize server health checks to verify content accessibility in large Web sites.
- Chapter 11, “High Availability,” describes how to keep your network available despite the failure of any single device. This high-availability network topology is supported through an enhanced implementation of the Virtual Router Redundancy Protocol (VRRP).

Part 3: Advanced Web Switching

- Chapter 12, “Global Server Load Balancing,” provides a configuration example for performing GSLB across multiple geographic sites. This chapter also describes how to configure client proximity tables to customize site preferences for client networks.
- Chapter 13, “Firewall Load Balancing,” describes how to use Web switches to divide the load between many discrete firewalls, allowing multiple active firewalls to operate in parallel.

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. Do not type the brackets.	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 covered by a maintenance contract.

- E-mail access:

  `support@alteon.com`

  E-mail access to Alteon WebSystems Customer Support is available for accounts 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.

Part 1: Basic Switching and Routing

This section discusses basic Layer 1 through 3 switching and routing functions. In addition to routing traffic at pure Ethernet speeds, Layer 3 switching was designed for multi-protocol routing.

- Basic IP Routing
- VLANs
- Port Trunking
- OSPF
- Border Gateway Protocol
- Secure Switch Management

050156A, June 2001

Alteon WebSystems
Web Speed for e-Business

CHAPTER 1

Basic IP Routing

This chapter provides configuration background and examples for using the switch to perform basic IP routing functions.

- "IP Routing Benefits" on page 18. This section briefly describes how IP routing enhances Alteon WebSystems switching solution.
- "Routing Between IP Subnets" on page 18. This section discusses Layer 2 switching and Layer 3 IP routing.
- "Example of Subnet Routing" on page 21. This section provides a step-by-step procedure on how to configure IP interfaces and add VLANs.
- "Defining IP Address Ranges for the Local Route Cache" on page 25. This section describes how to use your switch efficiently by configuring a local route cache.
- "DHCP Relay" on page 26. This section explains how you can reduce the number of DHCP servers deployed on the network and centralize them.

IP Routing Benefits

IP routing allows the network administrator to connect the server IP subnets to the rest of the backbone network, using a combination of configurable IP switch interfaces and IP routing options.

The IP routing feature enhances Alteon WebSystems switching solution in the following ways:

- Performs server load balancing (using both Layer 3 and Layer 4 switching in combination) to server subnets that are separate from backbone subnets
- Provides the ability to seamlessly route IP traffic between multiple Virtual Local Area Networks (VLANs) configured on the switch

Routing Between IP Subnets

The physical layout of most corporate networks has evolved over time. Classic hub/router topologies have given way to faster switched topologies, particularly now that switches are increasingly intelligent. Alteon 708s are intelligent and fast enough to perform routing functions on a par with wire speed Layer 2 switching.

The combination of faster routing and switching in a single device provides another service: it allows you to build versatile topologies that account for legacy configurations.

For example, consider the following topology migration:

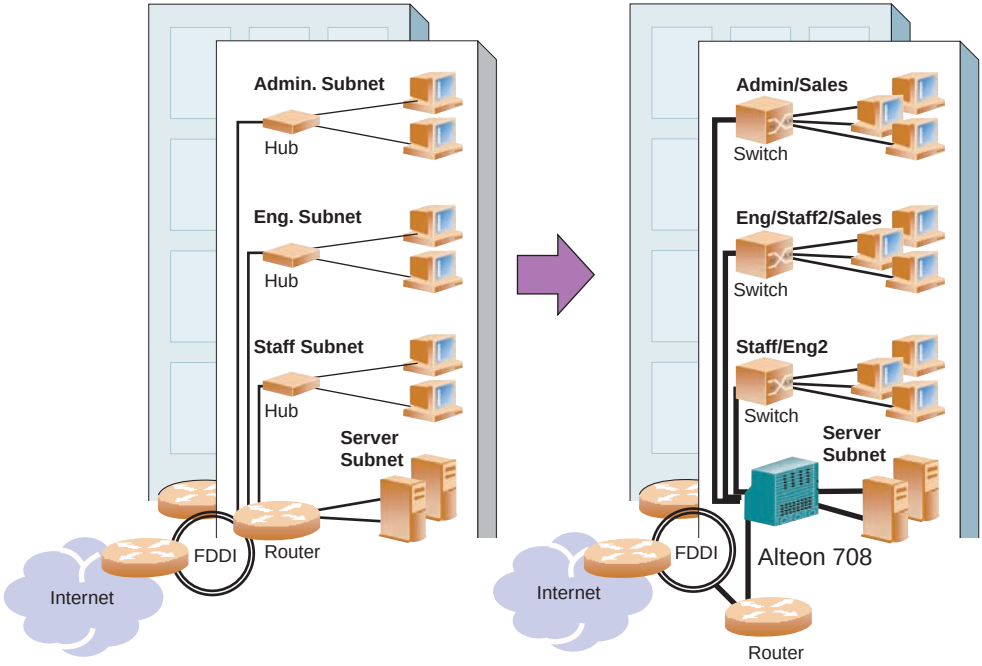


Figure 1-1 The Router Legacy Network

In this example, a corporate campus has migrated from a router-centric topology to a faster, more powerful, switch-based topology. As is often the case, the legacy of network growth and redesign has left the system with a jumble of illogically distributed subnets—a situation that switching alone cannot cure. Instead, the router is flooded with cross-subnet communication. This compromises efficiency in two ways:

- Routers can be slower than switches. The cross-subnet side trip from the switch to the router and back again adds two hops for the data, slowing throughput considerably.
- Traffic to the router increases, worsening any congestion.

Even if every end-station on the network could be moved to better logical subnets (a daunting task), competition for access to common server pools on different subnets still burdens the routers.

This problem is solved by using Alteon Web switches with built-in IP routing capabilities. Cross-subnet LAN traffic can now be routed within the Web OS-powered switches with wire speed Layer 2 switching performance. This not only eases the load on the router but saves the network administrators from reconfiguring each and every end-station with new IP addresses.

Take a closer look at the Alteon 708 in the following configuration example:

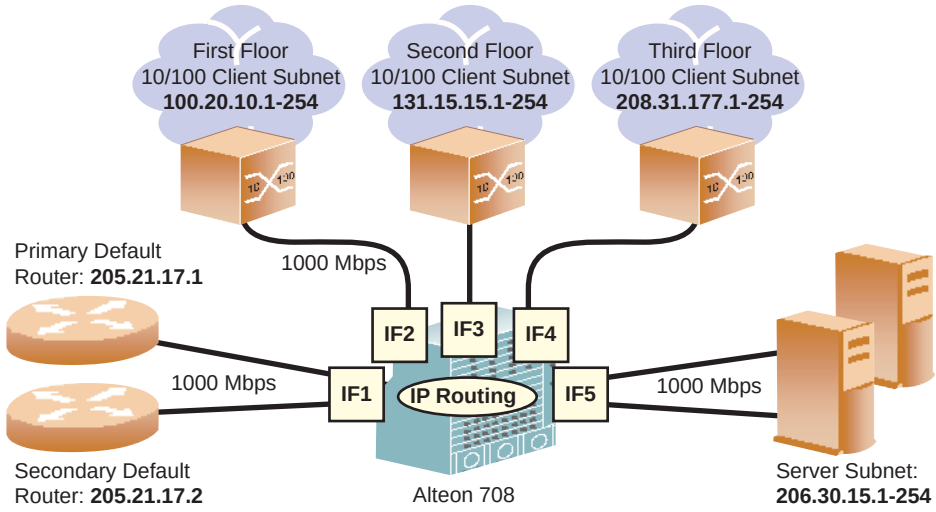


Figure 1-2 Switch-Based Routing Topology

The Alteon 708 connects the Gigabit Ethernet and Fast Ethernet trunks from various switched subnets throughout one building. Common servers are placed on another subnet attached to the switch. A primary and backup router are attached to the switch on yet another subnet.

Without Layer 3 IP routing on the switch, cross-subnet communication is relayed to the default gateway (in this case, the router) for the next level of routing intelligence. The router fills in the necessary address information and sends the data back to the switch. The switch then relays the packet to the proper destination subnet using Layer 2 switching.

With Layer 3 IP routing in place on the Alteon 708, routing between different IP subnets can be accomplished entirely within the switch. This leaves the routers free to handle inbound and outbound traffic for this group of subnets.

Example of Subnet Routing

Prior to configuration, you must be connected to the switch Command Line Interface (CLI) as the administrator (see Chapter 1 in the *Web OS 7.2 Command Reference*).

NOTE – For details about any of the menu commands described in this example, see "IP Configuration" in Chapter 6 of the *Web OS 7.2 Command Reference*.

1. **Assign an IP address (or document the existing one) for each real server, router, and client workstation.**

   In the example topology in Figure 1-2 on page 20, the following IP addresses are used:

Table 1-1 Subnet Routing Example: IP Address Assignments

Subnet	Devices	IP Addresses
1	Primary and Secondary Default Routers	205.21.17.1 and 205.21.17.2
2	First Floor Client Workstations	100.20.10.1-254
3	Second Floor Client Workstations	131.15.15.1-254
4	Third Floor Client Workstations	208.31.177.1-254
5	Common Servers	206.30.15.1-254

2. **Assign an IP interface for each subnet attached to the switch.**

   Since there are five IP subnets connected to the switch, five IP interfaces are needed:

Table 1-2 Subnet Routing Example: IP Interface Assignments

Interface	Devices	IP Interface Address
IF 1	Primary and Secondary Default Routers	205.21.17.3
IF 2	First Floor Client Workstations	100.20.10.16
IF 3	Second Floor Client Workstations	131.15.15.1
IF 4	Third Floor Client Workstations	208.31.177.2
IF 5	Common Servers	206.30.15.200

IP interfaces are configured using the following commands at the CLI:

```
>> Main# /cfg/ip/if 1                    (Select IP interface 1)
>> IP Interface 1# addr 205.21.17.3      (Assign IP address for the interface)
>> IP Interface 1# ena                   (Enable IP interface 1)
>> IP Interface 1# ../if 2               (Select IP interface 2)
>> IP Interface 2# addr 100.20.10.16     (Assign IP address for the interface)
>> IP Interface 2# ena                   (Enable IP interface 2)
>> IP Interface 2# ../if 3               (Select IP interface 3)
>> IP Interface 3# addr 131.15.15.1      (Assign IP address for the interface)
>> IP Interface 3# ena                   (Enable IP interface 3)
>> IP Interface 3# ../if 4               (Select IP interface 4)
>> IP Interface 4# addr 208.31.177.2     (Assign IP address for the interface)
>> IP Interface 4# ena                   (Enable IP interface 4)
>> IP Interface 4# ../if 5               (Select IP interface 5)
>> IP Interface 5# addr 206.30.15.200    (Assign IP address for the interface)
>> IP Interface 5# ena                   (Enable IP interface 5)
```

3. **Set each server and workstation's default gateway to the appropriate switch IP interface (the one in the same subnet as the server or workstation).**

4. **Configure the default gateways to the routers.**

   Configuring the default gateways allows the switch to send outbound traffic to the routers:

```
>> IP Interface 5# ../gw 1               (Select primary default gateway)
>> Default gateway 1# addr 205.21.17.1   (Assign IP address for primary router)
>> Default gateway 1# ena                (Enable primary default gateway)
>> Default gateway 1# ../gw 2            (Select secondary default gateway)
>> Default gateway 2# addr 205.21.17.2   (Set IP address for secondary router)
>> Default gateway 2# ena                (Enable secondary default gateway)
```

5. **Make sure IP forwarding is enabled.**

   IP forwarding is enabled by default. However, verify and make sure IP forwarding is enabled.

```
>> Default gateway 2# ../fwrd            (Select the IP Forwarding menu)
>> IP Forwarding # apply                 (Make your changes active)
>> IP Forwarding # /cfg/ip/cur           (View current IP settings)
```

6. **Save your new configuration changes.**

```
>> IP# save                              (Save for restore after reboot)
```

Adding VLANs to the Routing Example

In the previous example, devices that share a common IP network are all in the same broadcast domain. If you want to limit broadcasts on your network, you could use VLANs to create distinct broadcast domains. For example, you could create one VLAN for the routers, one for the servers, and one for the client trunks.

In this example, you are adding to the previous configuration.

1. **Determine which switch ports and IP interfaces belong to which VLANs.**

   The following table adds ports and VLANs information:

Table 1-3 Subnet Routing Example: Optional VLAN Ports

VLAN	Devices	IP Interface	Switch Port
1	First Floor Client Workstations	2	b1
	Second Floor Client Workstations	3	b2
	Third Floor Client Workstations	4	b3
2	Primary Default Router	1	b4
	Secondary Default Router	1	b5
3	Common Servers 1	5	b6
	Common Servers 2	5	b7

2. **Add the following ports to the respective VLANs.**

```
>> # /cfg/vlan 1                    (Select VLAN 1)
>> VLAN 1# add port b1              (Add port for 1st floor to VLAN 1)
>> VLAN 1# add port b2              (Add port for 2nd floor to VLAN 1)
>> VLAN 1# add port b3              (Add port for 3rd floor to VLAN 1)
>> VLAN 1# ../VLAN 2                (Select VLAN 2)
>> VLAN 2# add port b4              (Add port for default router 1)
>> VLAN 2# add port b5              (Add port for default router 2)
>> VLAN 2# ../VLAN 3                (Select VLAN 3)
>> VLAN 3# add port b6              (Select port for common server 1)
>> VLAN 3# add port b7              (Select port for common server 2)
```

Each time you add a port to a VLAN, you may get the following prompt:

```
Port B4 is untagged and VLAN 2 is not a configured PVID for port B4.
Would you like to change all PVIDs for port B4 to VLAN 2 [y|n]?
```

Enter **y** to set the default port VLAN ID (PVID) for the port.

3. **Enable the VLANs.**

```
>> Port B7# /cfg/vlan 1                    (Select VLAN 1, the client VLAN)
>> VLAN 1# ena                             (Enable VLAN 1)
>> VLAN 1# ../vlan 2                       (Select VLAN 2, the def. router VLAN)
>> VLAN 2# ena                             (Enable VLAN 2)
>> VLAN 2# ../vlan 3                       (Select VLAN 3, the server VLAN)
>> VLAN 3# ena                             (Enable VLAN 3)
```

4. **Add each IP interface to the appropriate VLAN.**

Now that the ports are separated into three VLANs, the IP interface for each subnet must be placed in the appropriate VLAN. From Table 1-3 on page 23, the settings are made as follows:

```
>> VLAN 3# /cfg/ip/if 1                    (Select IP interface 1 for def. routers)
>> IP Interface 1# vlan 2                  (Set to VLAN 2)
>> IP Interface 1# ../if 2                 (Select IP interface 2 for first floor)
>> IP Interface 2# vlan 1                  (Set to VLAN 1)
>> IP Interface 2# ../if 3                 (Select IP interface 3 for second floor)
>> IP Interface 3# vlan 1                  (Set to VLAN 1)
>> IP Interface 3# ../if 4                 (Select IP interface 4 for third floor)
>> IP Interface 4# vlan 1                  (Set to VLAN 1)
>> IP Interface 4# ../if 5                 (Select IP interface 5 for servers)
>> IP Interface 5# vlan 3                  (Set to VLAN 3)
```

5. **Apply and verify the configuration.**

```
>> IP Interface 5# apply                   (Make your changes active)
>> IP Interface 5# /info/vlan              (View current VLAN information)
>> Information# port                       (View current port information)
```

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

6. **Save your new configuration changes.**

```
>> Information# save                       (Save for restore after reboot)
```

Defining IP Address Ranges for the Local Route Cache

A local route cache lets you use switch resources more efficiently. The local network address and local network mask parameters (accessed via the `/cfg/ip/frwd/local/add` command) define a range of addresses that will be cached on the switch. The *local network address* is used to define the base IP address in the range that will be cached. The *local network mask* is applied to produce the range. To determine if a route should be added to the memory cache, the destination address is masked (bit-wise AND) with the local network mask and checked against the local network address.

By default, the local network address and local network mask are both set to 0.0.0.0. This produces a range that includes all Internet addresses for route caching: 0.0.0.0 through 255.255.255.255.

To limit the route cache to your local hosts, you could configure the parameters as shown in the following example:

Table 1-4 Local Routing Cache Address Ranges

Local Host Address Range	Local Network Address	Local Network Mask
0.0.0.0 - 127.255.255.255	0.0.0.0	128.0.0.0
128.0.0.0 - 128.255.255.255	128.0.0.0	128.0.0.0 or 255.0.0.0
205.32.0.0 - 205.32.255.255	205.32.0.0	255.255.0.0

NOTE – All addresses that fall outside the defined range are forwarded to the default gateway. The default gateways must be within the configured range.

DHCP Relay

Dynamic Host Configuration Protocol (DHCP) is a transport protocol that provides a framework for automatically assigning IP addresses and configuration information to other IP hosts or clients in a large TCP/IP network. WIthout DHCP, the IP address must be entered manually for each network device. DHCP allows a network administrator to distribute IP addresses from a central point and automatically send a new IP address when a device is connected to a different place in the network.

DHCP is an extension of another network IP management protocol, Bootstrap Protocol (BOOTP), with an additional capability of being able to dynamically allocate resuable network addresses and configuration parameters for client operation.

Built on the client/server model, DHCP allows hosts or clients on an IP network to obtain their configurations from a DHCP server, thereby reducing network administration.The most significant configuration the client receives from the server is its required IP address; (other optional parameters include the “generic” file name to be booted, the address of the default gateway, and so forth).

Alteon WebSystems DHCP relay agent eliminates the need to have DHCP/BOOTP servers on every subnet. It allows the administrator to reduce the number of DHCP servers deployed on the network and to centralize them. Without the DHCP relay agent, there must be at least one DHCP server deployed at each subnet that has hosts needing to perform the DHCP request.

DHCP Overview

DHCP is described in RFC 2131, and the DHCP relay agent supported on Alteon 708 is described in RFC 1542. DHCP uses UDP as its transport protocol. The client sends messages to the server on Port 67, and the server sends messages to the client on Port 68.

DHCP defines the methods through which clients can be assigned an IPaddress for a finite lease period and allowing reassignment of the IP address to another client later. Additionally, DHCP provides the mechanism for a client to gather other IP configuration parameters it needs to operate in the TCP/IP network.

In the DHCP environment, the Alteon Web switch acts as a relay agent. The DHCP relay feature (`/cfg/ip/bootp`) enables the switch to forward a client request for an IP address to two BOOTP servers with IP addresses that have been configured on the switch.

When a switch receives a UDP broadcast on port 67 from a DHCP client requesting an IP address, the switch acts as a proxy for the client, replacing the client source IP (SIP) address and destination IP (DIP) address. The request is then forwarded as a UDP Unicast MAC layer message to two BOOTP servers whose IP addresses are configured on the switch.

The servers respond as a UDP Unicast message back to the switch, with the default gateway and IP address for the client. The destination IP address in the server response represents the interface address on the switch that received the client request. This interface address tells the switch on which VLAN to send the server response to the client.

Configuring a DHCP Relay Agent

To enable the Alteon 708 to be the BOOTP forwarder, you need to configure the DHCP/BOOTP server IP addresses on the switch. Generally, you should configure the command on the interface closest to the client, so that the DHCP server knows from which IP subnet the newly allocated IP address should come.

The following figure shows a basic DHCP network example.

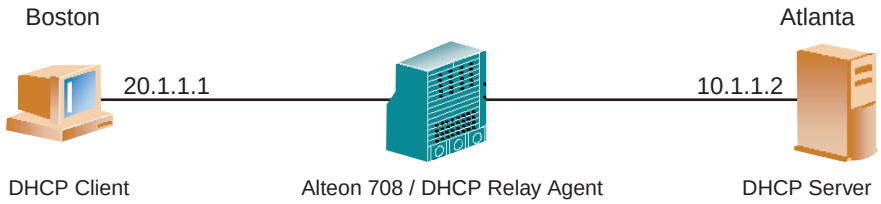


Figure 1-3 DHCP Relay Agent Configuration

With the Alteon 708, there is no need for primary or secondary servers. The client request is simply forwarded to the BOOTP servers configured on the switch. The use of two BOOTP servers provides failover redundancy. However, no health checking is supported.

Use the following commands to configure the switch as a DHCP relay agent.

```
>> /cfg/ip/bootp
>> Bootstrap Protocol Relay# addr     (Set IP address of BOOTP server)
>> Bootstrap Protocol Relay# addr2    (Set IP address of 2nd BOOTP server)
>> Bootstrap Protocol Relay# on       (Globally turn BOOTP relay ON )
>> Bootstrap Protocol Relay# off      (Globally turn BOOTP relay OFF)
>> Bootstrap Protocol Relay# cur      (Display current BOOTP relay configuration)
```

Additionally, DHCP Relay functionality can be assigned on a per interface basis. Use the following command to enable the Relay functionality:

```
/cfg/ip/if <interface number>/relay ena
```

CHAPTER 2
VLANs

This chapter describes network design and topology considerations for using Virtual Local Area Networks (VLANs). VLANs are commonly used to split up groups of network users into manageable broadcast domains, to create logical segmentation of workgroups, and to enforce security policies among logical segments. The following topics are discussed in this chapter:

- "VLAN ID Numbers" on page 30. This section briefly describes the concept of VLAN ID numbers.
- "VLAN Tagging" on page 30. This section discusses 802.1Q VLAN *tagging* that provides standards-based VLAN support for Ethernet systems.
- "VLANs and the IP Interfaces" on page 31. This section briefly describes how management functions can only be accomplished from stations on VLANs that include an IP interface to the switch.
- "VLAN Topologies and Design Issues" on page 31. This section discusses how you can logically connect users and segments to a host that supports many logical segments or subnets using the flexibility of the multiple VLAN system.
- "VLANs and Multiple Spanning Trees" on page 35. This section discusses how Spanning Tree Protocol (STP) forces redundant data paths into a standby (blocked) state. Web OS supports 256 multiple spanning trees.

NOTE – Basic VLANs can be configured during initial switch configuration (see "Using the Setup Utility" in the Web OS 7.2 Command Reference). More comprehensive VLAN configuration can be done from the Command Line Interface (CLI) (see "VLAN Configuration" as well as "Port Configuration" in the Web OS 7.2 Command Reference).

VLAN ID Numbers

Web OS supports up to 1024 VLANs per switch. Even though the maximum number of VLANs supported at any given time is 1024, each can be identified with any number between 1–4094.

VLANs are defined on a per-port basis. Each port on the switch can belong to one or more VLANs, and each VLAN can have any number of switch ports in its membership. Any port that belongs to multiple VLANs, however, must have VLAN *tagging* enabled (see the following section).

Each port in the switch has a configurable default VLAN number, known as its *PVID*. Each port can have multiple PVIDs based on the ethertype. The factory default value of all PVIDs is 1. This places all ports on the same VLAN initially, although each port's PVID is configurable to any VLAN number between 1– 4094.

Any non-tagged frames (those with no VLAN specified) are classified with the sending port's PVID for the specific ethertype.

VLAN Tagging

Web OS software supports 802.1Q VLAN *tagging*, providing standards-based VLAN support for Ethernet systems.

Tagging places the VLAN identifier in the frame header, allowing multiple VLANs per port. When you configure multiple VLANs on a port, you must also enable tagging on that port.

Since tagging fundamentally changes the format of frames transmitted on a tagged port, you must carefully plan network designs to prevent tagged frames from being transmitted to devices that do not support 802.1Q VLAN tags.

VLANs and the IP Interfaces

Careful consideration must be made when creating VLANs within the switch, such that communication with the switch Management Processor (MP) remains possible where it is required.

Access to the switch for remote configuration, trap messages, and other management functions can only be accomplished from stations on VLANs that include an IP interface to the switch (see “IP Interface Menu” in Chapter 6 of the *Web OS 7.2 Command Reference*). Likewise, access to management functions can be cut off to any VLAN by excluding IP interfaces from its membership.

For example, if all IP interfaces are left on VLAN 1 (the default), and all ports are configured for VLANs other than VLAN 1, then switch management features are effectively cut off. If an IP interface is added to one of the other VLANs, the stations in that VLAN all have access to switch management features.

VLAN Topologies and Design Issues

By default, the Web OS software has a single VLAN configured on every port. This configuration groups all ports into the same broadcast domain. The VLAN has an 802.1Q VLAN PVID of 1. VLAN tagging is turned off, because by default only a single VLAN is configured per port.

Since VLANs are most commonly used to create individual broadcast domains and/or separate IP subnets, it is useful for host systems to be able to have presence on more than one VLAN simultaneously. Alteon 708 switches and VLAN-tagging adapters have the unique capability of being able to support multiple VLANS on a per port or per interface basis, allowing very flexible configurations.

You can configure multiple VLANs on a single adapter on a server, with each VLAN being configured through a logical interface and logical IP address on the host system. Each VLAN configured on the adapter must also be configured on the switch port to which it is connected. If multiple VLANs are configured on the port, tagging must be turned on.

Using this flexible multiple VLAN system, you can logically connect users and segments to a host with a single VLAN-tagging adapter that supports many logical segments or subnets.

Example 1: Multiple VLANS with Tagging Adapters

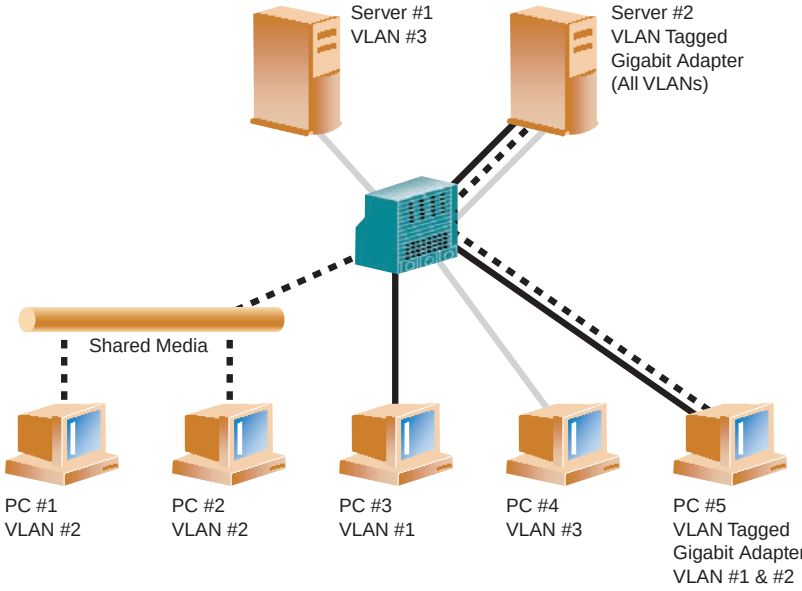


Figure 2-1 Example 1: Multiple VLAN-Tagged Adapters

The features of this VLAN are described below:

Table 2-1 Example VLAN Topology

Component	Description
Alteon 708	This switch is configured for three VLANs that represent three different IP subnets. Two servers and five clients are attached to the switch.
Server 1	This server is part of the VLAN 3 and only has presence in one IP subnet. The port to which the VLAN is attached is configured for only VLAN 3, so VLAN tagging is off.
Server 2	This high-use server needs to be accessed from all VLANs and IP subnets. The server has a multiple VLAN-tagging adapter installed with VLAN tagging turned on. The adapter is attached to one of the Alteon 708 switch's Gigabit Ethernet ports, that is configured for VLANs 1, 2, and 3. Tagging is turned on. Because of the VLAN-tagging capabilities of both the adapter and the switch, the server is able to communicate on all three IP subnets in this network. Broadcast separation between all three VLANs and subnets, however, is maintained.
PCs 1 and 2	These PCs are attached to a shared media hub that is then connected to the switch. They belong to VLAN 2 and are logically in the same IP subnet as Server 2 and PC 5. Tagging is not enabled on their switch port.

Table 2-1 Example VLAN Topology

Component	Description
PC 3	A member of VLAN 1, this PC can only communicate with Server 2 and PC 5.
PC 4	A member of VLAN 3, this PC can only communicate with Server 1 and Server 2.
PC 5	A member of both VLAN 1 and VLAN 2, this PC has a VLAN-tagged Gigabit Ethernet Adapter installed. It is able to communicate with Server 2 via VLAN 1 and to PC 1 and PC 2 via VLAN 2. The switch port to which it is connected is configured for both VLAN 1 and VLAN 2 and has tagging turned on.

NOTE – VLAN tagging is only required on ports that are connected to other Alteon WebSystems switches or on ports that connect to tag-capable end-stations, such as servers with VLAN-tagged Gigabit Ethernet Adapters.

Example 2: Parallel Links with VLANs

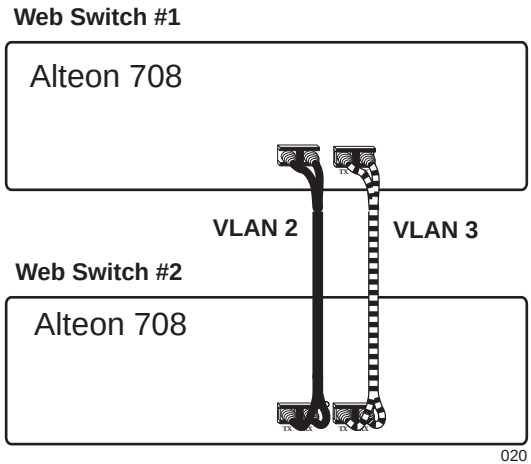


Figure 2-2 Example 2: Parallel Links with VLANs

The following items describe the features of this example:

- Figure 2-2 on page 34 shows how to use VLANs to create configurations with multiple links between two switches without creating broadcast loops.
- Two Alteon 708 switches are connected with two different Gigabit Ethernet links. Without VLANs, this configuration would create a broadcast loop; however, the Spanning Tree Protocol (STP) topology resolves parallel loop-creating links.

NOTE – To prevent broadcast loops, configure interswitch links on separate VLANs, as shown in Figure 2-2.

- With VLANs, no switch-to-switch link shares the same VLAN and, thus, the VLANs are separated into their own broadcast domains.

VLANs and Multiple Spanning Trees

When Spanning Tree Protocol (STP) is enabled on the switch, it detects and eliminates logical loops in a bridged or switched network. STP forces redundant data paths into a standby (blocked) state. When multiple paths exist, Spanning Tree configures the network so that a switch uses only the most efficient path. If that path fails, Spanning Tree automatically sets up another active path on the network to sustain network operations.

If you do not use Spanning Tree, you can avoid the lengthy delays that can occur (the protocol's sequence of listening, learning, then forwarding or blocking) while STP is resolving itself. Instead of enabling Spanning Tree, you can use VLANs.

Web OS supports up to 256 multiple spanning trees. A Spanning Tree can have one VLAN only, except for the default Spanning Tree group, which can have one or more VLANs. You can enable or disable a Spanning Tree group on a per port basis, but the Alteon 708 cannot enable different instances of a Spanning Tree group per VLAN.

In the factory default configuration, all spanning tree groups are empty of VLANs except for Spanning Tree Group 1. VLAN 1 defaults to Spanning Tree 1 membership. As you create additional VLANs, the new VLANs belong to Spanning Tree Group 1 by default unless you select another Spanning Tree.

Port, trunk groups, VLANs, and Spanning Trees are related to each other, as shown in Table 2-2.

NOTE – A port cannot belong to a trunk group and a VLAN.

Table 2-2 Ports, Trunk Groups, and VLANs

Switch Elements	Belongs to
Port	–Trunk group –One or more VLANs
Trunk group	One or more VLANs
VLAN	One Spanning Tree group

NOTE – Spanning Tree is enabled by default on VLAN 1 and on all newly created VLANs.

Typically, when you implement multiple Spanning Trees at customer sites, each site is given a Spanning Tree to eliminate logical loops. Figure 2-3 shows two Spanning Tree groups configured for three VLANs.

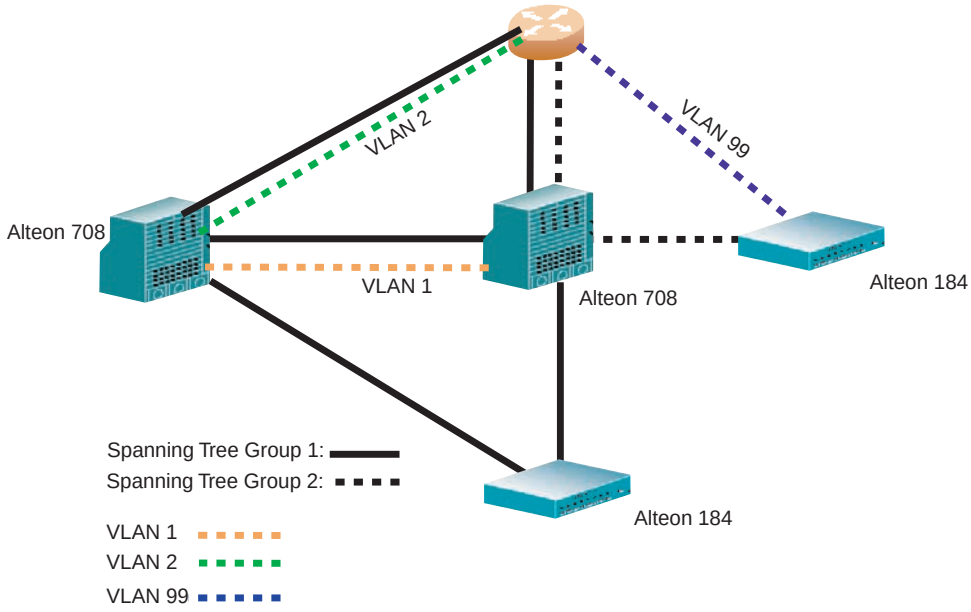


Figure 2-3 Multiple Spanning Trees Across Alteon 708s

VLAN 1 and VLAN 2 belong to the default Spanning Tree Group 1 and VLAN 99 belongs to Spanning Tree Group 2. VLANs cannot be shared between Spanning Trees. The above configuration shows both the Alteon 708s supporting multiple Spanning Trees, but Alteon 184s cannot belong to the same Spanning Tree group.

Depending on your hardware, you can enable Spanning Tree on a per-VLAN or a global basis. In either case, the switch maintains a separate instance of Spanning Tree for each VLAN, except on VLANs where you disable Spanning Tree.

CHAPTER 3

Port Trunking

This chapter provides configuration background and examples for trunking multiple ports together in the following sections:

- "Port Trunking Overview" on page 38. This section discusses the basics of port trunking, traffic distribution in trunk groups, and fault tolerance.
- "Port Trunking Example" on page 40. This section provides a step-by-step procedure to configure an Alteon 708 for port trunking.

Port Trunking Overview

Trunk groups can provide super-bandwidth, multi-link connections between Alteon Web switches or other trunk-capable devices. A trunk group is a group of ports that act together, combining their bandwidth to create a single, larger virtual link.

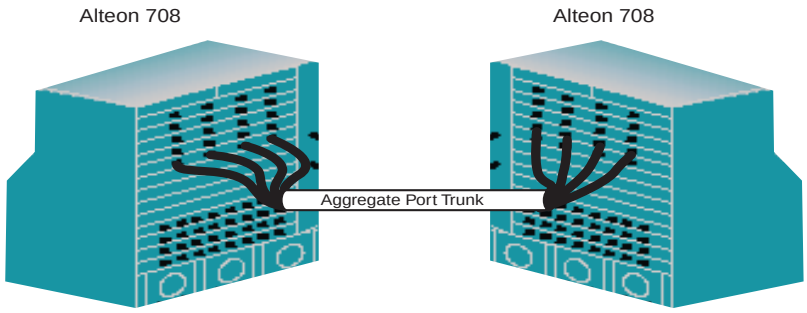


Figure 3-1 Gigabit Port Trunk Group

When using port trunk groups between two Alteon 708 switches, the network administrator can create a virtual link between the switches operating up to six gigabits per second, depending on how many physical ports are combined. The switch supports up to four trunk groups per switch, each with two to four links.

Trunk groups are also useful for connecting an Alteon WebSystems switch to third-party devices that support link aggregation, such as Cisco routers and switches with EtherChannel technology (*not* ISL trunking technology), and Sun's Quad Fast Ethernet Adapter. Alteon WebSystems' trunk group technology is compatible with these devices when they are configured manually.

Statistical Load Distribution

Network traffic is statistically load balanced between the ports in a trunk group. The Web OS-powered switch uses both the Layer 2 MAC address and Layer 3 IP address information present in each transmitted frame for determining load distribution.

The addition of Layer 3 IP address examination is an important advance for traffic distribution in trunk groups. In some port trunking systems, only Layer 2 MAC addresses are considered in the distribution algorithm. Each packet's particular combination of source and destination MAC addresses results in selecting one line in the trunk group for data transmission. If there are enough Layer 2 devices feeding the trunk lines, then traffic distribution becomes relatively even.

In some topologies, however, only a limited number of Layer 2 devices (such as a handful of routers and servers) feed the trunk lines. When this occurs, the limited number of MAC address combinations encountered results in lopsided traffic distribution, which can reduce the effective combined bandwidth of the trunked ports.

By adding Layer 3 IP address information to the distribution algorithm, a far wider variety of address combinations is available. Even with just a few routers feeding the trunk, the normal source/destination IP address combinations (even within a single LAN) can be widely varied. This results in a wider statistical load distribution and maximizes the use of the combined bandwidth available to trunked ports.

Built-In Fault Tolerance

Since each trunk group is comprised of multiple physical links, the trunk group is inherently fault tolerant. As long as one connection between the switches is available, the trunk remains active.

Statistical load balancing is also maintained even if a port in a trunk group is lost or returned to service.

Port Trunking Example

In this example, four gigabit ethernet ports are trunked between two Alteon 708s.

Prior to configuring each switch in this example, you must connect to the appropriate switch's Command Line Interface (CLI) as the administrator (see the *Web OS 7.2 Command Reference*).

NOTE – For details about any of the menu commands described in this example, see "Trunk Configuration" in Chapter 6 of the *Web OS 7.2 Command Reference*.

1. **Connect the switch ports that will be involved in the trunk group.**

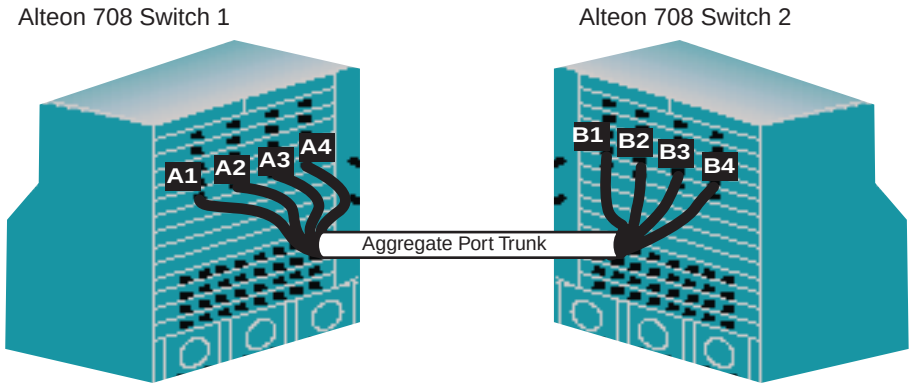


Figure 3-2 Configuring a Port Trunk Group

2. **On Switch 1, define a trunk group.**

```
>> Main # /cfg/trunk 1                    (Select trunk group 1)
>> Trunk group 1 # add a1                 (Add port a1 to trunk group 1)
>> Trunk group 1 # add a2                 (Add port a2 to trunk group 1)
>> Trunk group 1 # add a3                 (Add port a3 to trunk group 1)
>> Trunk group 1 # add a4                 (Add port a4 to trunk group 1)
>> Trunk group 1 # ena                    (Enable trunk group 1)
```

3. **On Switch 1, apply and verify the configuration.**

```
>> Trunk group 1 # apply                  (Make your changes active)
>> Trunk group 1 # cur                    (View current trunking configuration)
```

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

4. **On Switch 1, save your new configuration changes.**

```
>> Trunk group 1 # save                    (Save for restore after reboot)
```

5. **On Switch 2, repeat the process.**

```
>> Main # /cfg/trunk 3                     (Select trunk group 3)
>> Trunk group 3 # add b1                  (Add port b1 to trunk group 3)
>> Trunk group 3 # add b2                  (Add port b2 to trunk group 3)
>> Trunk group 3 # add b3                  (Add port b3 to trunk group 3)
>> Trunk group 3 # add b4                  (Add port b4 to trunk group 3)
>> Trunk group 3 # ena                     (Enable trunk group 3)
>> Trunk group 3 # apply                   (Make your changes active)
>> Trunk group 3 # cur                     (View current trunking configuration)
>> Trunk group 3 # save                    (Save for restore after reboot)
```

Trunk group 1 (on Switch 1) is now connected to trunk group 3 (on Switch 2).

NOTE – In this example, two Alteon WebSystems switches are used. If using a third-party device that supports link aggregation (such as Cisco routers and switches with EtherChannel technology or Sun's Quad Fast Ethernet Adapter), configure trunk groups on the third-party device manually. Connection problems could arise when using automatic trunk group negotiation on the third-party device.

---

6. **Examine the trunking information on each switch.**

```
>> /info/trunk                             (View trunking information)
```

Information about each port in each configured trunk group will be displayed. Make sure that trunk groups consist of the expected ports and that each port is in the expected state.

The following restrictions apply:

- Any physical switch port can belong to only one trunk group.
- Up to four ports can belong to the same trunk group.
- Best performance is achieved when all ports in any given trunk group are configured for the same speed.
- Trunking from non-Alteon WebSystems devices must comply with Cisco® EtherChannel® technology.

CHAPTER 4
OSPF

Web OS 7.2 supports the Open Shortest Path First (OSPF) routing protocol. Alteon WebSystems' implementation conforms to the OSPF version 2 specifications detailed in Internet RFC 1583. The following sections discuss OSPF support for the Alteon 708 Web switch:

- "OSPF Overview" on page 44. This section provides information on OSPF concepts: Types of OSPF areas, types of routing devices, neighbors, adjacencies, link state database, authentication, and internal versus external routing.
- "Web OS 7.2 OSPF Implementation" on page 48. This section gives you information specific to OSPF 7.2 implementation: Configuration parameters, electing the designated router, summarizing routes, defining route maps and so forth.
- "OSPF Configuration Examples" on page 59. This section provides step-by-step instructions on configuring four different configuration examples:
  - Creating a simple OSPF domain
  - Creating virtual links
  - Summarizing routes
  - Creating host routes

OSPF Overview

OSPF is designed for routing traffic within a single IP domain called an Autonomous System (AS). The AS can be divided into smaller logical units known as *areas*.

All routing devices maintain link information in their own Link State Database (LSDB). The LSDB for all routing devices within an area is identical but is not exchanged between different areas. Only routing updates are exchanged between areas, thereby significantly reducing the overhead for maintaining routing information on a large, dynamic network.

The following sections describe key OSPF concepts.

Types of OSPF Areas

An AS can be broken into logical units known as *areas*. In any AS with multiple areas, one area must be designated as area 0, known as the *backbone*. The backbone acts as the central OSPF area. All other areas in the AS must be connected to the backbone. Areas inject summary routing information into the backbone, which then distributes it to other areas as needed.

As shown in Figure 4-1, OSPF defines the following types of areas:

- Stub Area—an area that is connected to only one other area. External route information is not distributed into stub areas.
- Not-So-Stubby-Area (NSSA)—similar to a stub area with additional capabilities. Routes originating from within the NSSA can be propagated to adjacent transit and backbone areas. External routes from outside the AS can be advertised within the NSSA but are not distributed into other areas.
- Transit Area—an area that allows area summary information to be exchanged between routing devices. The backbone (area 0), any area that contains a virtual link to connect two areas, and any area that is not a stub area or an NSSA are considered transit areas.

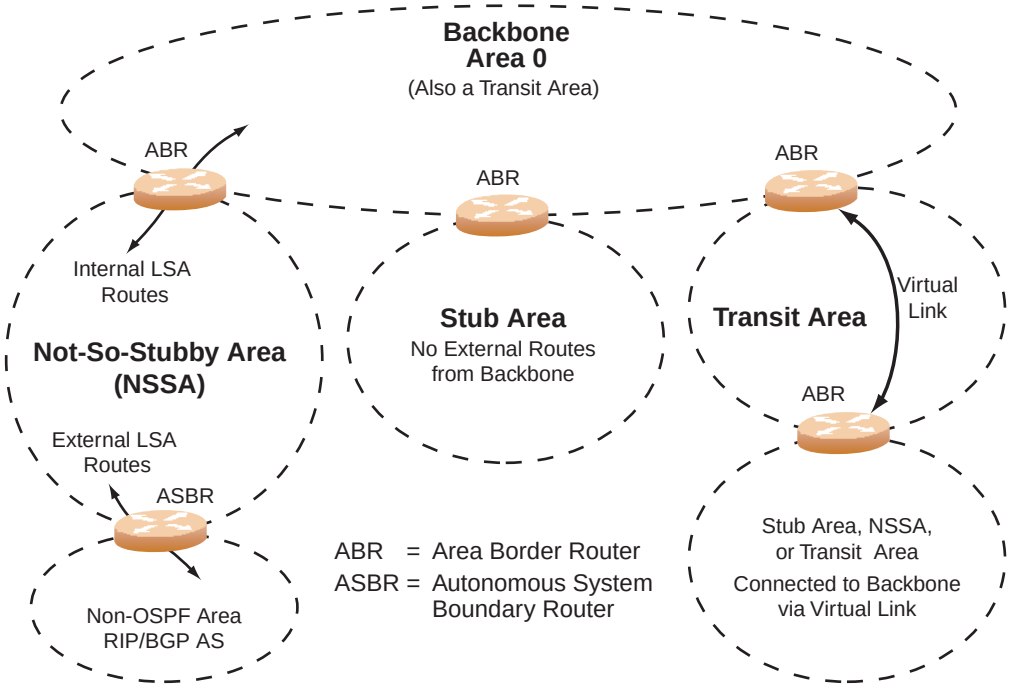


Figure 4-1 OSPF Area Types

Types of OSPF Routing Devices

As shown in Figure 4-2, OSPF uses the following types of routing devices:

- Internal Router (IR)—a router that has all of its interfaces within the same area. IRs maintain LSDBs identical to those of other routing devices within the local area.
- Area Border Router (ABR)—a router that has interfaces in multiple areas. ABRs maintain one LSDB for each connected area and disseminate routing information between areas.
- Autonomous System Boundary Router (ASBR)—a router that acts as a gateway between the OSPF domain and non-OSPF domains, such as RIP, BGP, and static routes.

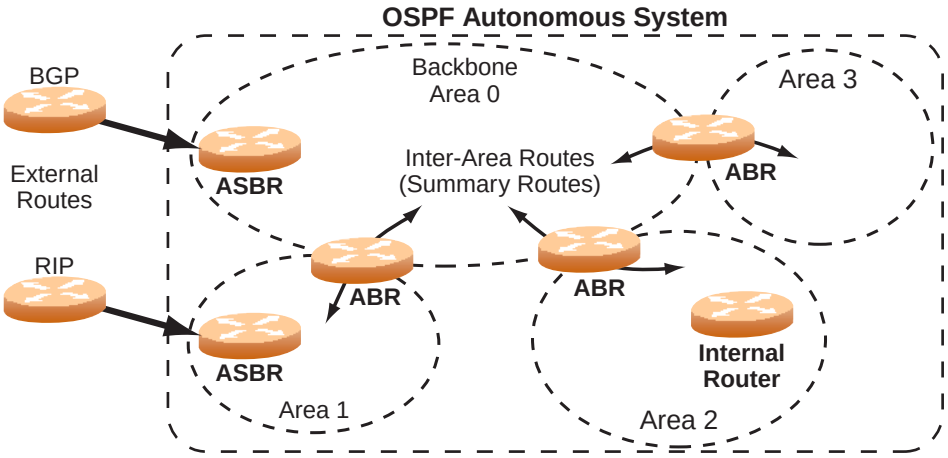


Figure 4-2 OSPF Domain and an Autonomous System

Neighbors and Adjacencies

In areas with two or more routing devices, *neighbors* and *adjacencies* are formed.

Neighbors are routing devices that maintain information about each others' health. To establish neighbor relationships, routing devices periodically send hello packets on each of their interfaces. All routing devices that share a common network segment, appear in the same area, and have the same health parameters (`hello` and `dead` intervals) and authentication parameters respond to each other's hello packets and become neighbors. Neighbors continue to send periodic hello packets to advertise their health to neighbors. In turn, they listen to hello packets to determine the health of their neighbors and to establish contact with new neighbors.

Adjacencies are neighbors that exchange OSPF database information. In order to limit the number of database exchanges, not all neighbors in an area (IP network) become adjacent to each other. Instead, the hello process is used for electing one of the neighbors as the area's Designated Router (DR) and one as the area's Backup Designated Router (BDR).

The DR is adjacent to all other neighbors and acts as the central contact for database exchanges. Each neighbor sends its database information to the DR, which relays the information to the other neighbors.

Because of the overhead required for establishing a new DR in case of failure, the hello process also elects a Backup Designated Router (BDR). The BDR is adjacent to all other neighbors (including the DR). Each neighbor sends its database information to the BDR just as with the DR, but the BDR merely stores this data and does not distribute it. If the DR fails, the BDR will take over the task of distributing database information to the other neighbors.

The Link-State Database

OSPF is a link-state routing protocol. A *link* represents an interface (or routable path) from the routing device. By establishing an adjacency with the DR, each routing device in an OSPF area maintains an identical Link-State Database (LSDB) describing the network topology for its area.

Each routing device transmits a Link-State Advertisement (LSA) on each of its interfaces. LSAs are entered into the LSDB of each routing device. OSPF uses *flooding* to distribute LSAs between routing devices.

When LSAs result in changes to the routing device's LSDB, the routing device forwards the changes to the adjacent neighbors (the DR and BDR) for distribution to the other neighbors.

OSPF routing updates occur only when changes occur, instead of periodically. For each new route, if an adjacency is interested in that route (for example, if configured to receive static routes and the new route is indeed static), an update message containing the new route is sent to the adjacency. For each route removed from the route table, if the route has already been sent to an adjacency, an update message containing the route to withdraw is sent.

The Shortest Path First Tree

The routing devices use a link-state algorithm (Dijkstra's algorithm) to calculate the shortest path to all known destinations, based on the cumulative *cost* required to reach the destination.

The cost of an individual interface in OSPF is an indication of the overhead required to send packets across it. The cost is inversely proportional to the bandwidth of the interface. A lower cost indicates a higher bandwidth.

Authentication

OSPF also allows packet authentication and uses IP multicast when sending and receiving packets. This ensures less processing on routing devices that are not listening to OSPF packets.

Internal Versus External Routing

To ensure effective processing of network traffic, every routing device on your network needs to know how to send a packet (directly or indirectly) to any other location/destination in your network. This is referred to as *internal routing* and can be done with static routes or using active internal routing protocols, such as OSPF, RIP, or RIPv2.

It is also useful to tell routers outside your network (upstream providers or *peers*) about the routes you have access to in your network. Sharing of routing information between autonomous systems is known as *external routing*.

Typically, an AS will have one or more border routers (peer routers that exchange routes with other OSPF networks) as well as an internal routing system enabling every router in that AS to reach every other router and destination within that AS.

When a routing device *advertises* routes to boundary routers on other autonomous systems, it is effectively committing to carry data to the IP space represented in the route being advertised. For example, if the routing device advertises 192.204.4.0/24, it is declaring that if another router sends data destined for any address in the 192.204.4.0/24 range, it will carry that data to its destination.

Web OS 7.2 OSPF Implementation

The following sections describe issues specific to the OSPF implementation in Web OS 7.2:

- "Configurable Parameters" on page 48
- "Defining Areas" on page 49
- "Interface Cost" on page 52
- "Electing the Designated Router and Backup" on page 52
- "Summarizing Routes" on page 52
- "Default Routes" on page 53
- "Virtual Links" on page 54
- "Router ID" on page 54
- "Authentication" on page 55
- "Host Routes for Load Balancing" on page 55
- "Redistributing Routes into OSPF" on page 56
- "OSPF Features Not Supported in This Release" on page 59

Configurable Parameters

In Web OS 7.2, OSPF parameters can be configured through the Command Line Interface (CLI), Web OS Browser-Based Interface (BBI), or through SNMP.

The CLI supports the following parameters: interface output cost, interface priority, dead and hello intervals, retransmission interval, and interface transmit delay.

In addition, you can specify the following:

- Shortest Path First (SPF) interval—Time interval between successive calculations of the shortest path tree using the Dijkstra's algorithm.
- Stub area metric—A stub area can be configured to send a numeric metric value such that all routes received via that stub area carry the configured metric to potentially influence routing decisions.
- Default routes—Default routes with weight metrics can be manually injected into transit areas. This helps establish a preferred route when multiple routing devices exist between two areas. It also helps route traffic to external networks.

Defining Areas

If you are configuring multiple areas in your OSPF domain, one of the areas must be designated as area 0, known as the *backbone*. The backbone is the central OSPF area and is usually physically connected to all other areas. The areas inject routing information into the backbone which, in turn, disseminates the information into other areas.

Since the backbone connects the areas in your network, it must be a contiguous area. If the backbone is partitioned (possibly as a result of joining separate OSPF networks), parts of the AS will be unreachable, and you will need to configure *virtual links* to reconnect the partitioned areas (see "Virtual Links" on page 54).

Up to eight OSPF areas can be connected to an Alteon 708 with Web OS 7.2 software. To configure an area, the OSPF number must be defined and then attached to a network interface on the Web switch. The full process is explained in the following sections.

An OSPF area is defined by assigning ***two*** pieces of information—an *area index* and an *area ID*. The command to define an OSPF area is as follows:

```
>> # /cfg/ip/ospf/aindex <area index>/areaid <area ID number>
```

NOTE – The `aindex` option above is an arbitrary index used only on the switch and does not represent the actual OSPF area number. The actual OSPF area number is defined in the `areaid` portion of the command as will be explained below.

Assigning the Area Index

The `aindex` <*area index*> option is actually just an arbitrary index (0-7) used only by the Web switch. This index does not necessarily represent the OSPF area number, though for configuration simplicity, it should where possible.

For example, both of the following sets of commands define OSPF area 0 (the backbone) and area 1 because that information is held in the area ID portion of the command. However, the first set of commands is easier to configure and maintain because the arbitrary area indexes agree with the area IDs:

- Area index and area ID agree

  `/cfg/ip/ospf/aindex 0/areaid 0.0.0.0` *(Use index 0 to set area 0 in ID octet format)*
  `/cfg/ip/ospf/aindex 1/areaid 0.0.0.1` *(Use index 1 to set area 1 in ID octet format)*

- Also valid, the area index can be set to an arbitrary value

  `/cfg/ip/ospf/aindex 4/areaid 0.0.0.0` *(Use index 4 to set area 0 in ID octet format)*
  `/cfg/ip/ospf/aindex 5/areaid 0.0.0.1` *(Use index 5 to set area 1 in ID octet format)*

Using the Area ID to Assign the OSPF Area Number

The OSPF area number is defined in the `areaid` <*IP address*> option. The octet format is used in order to be compatible with two different systems of notation used by other OSPF network vendors. There are two valid ways to designate an area ID:

- Placing the area number in the last octet (0.0.0.*n*)

  Most common OSPF vendors express the area ID number as a single number. For example, the Cisco IOS-based router command "`network 1.1.1.0 0.0.0.255 area 1`" defines the area number simply as "`area 1.`" On an Alteon 708 Web switch, using the last octet in the area ID, "`area 1`" is equivalent to "`areaid 0.0.0.1`".

- Multi-octet (*IP address*)

  Some OSPF vendors express the area ID number in multi-octet format. For example, "`area 2.2.2.2`" represents OSPF area 2 and can be specified directly on an Alteon 708 Web switch as "`areaid 2.2.2.2`".

NOTE – Although both types of area ID formats are supported, be sure that the area IDs are in the same format throughout an area.

Attaching an Area to a Network

Once an OSPF area has been defined, it must be associated with a network. To attach the area to a network, you must assign the OSPF area index to an IP interface that participates in the area. The format for the command is as follows:

```
>> # /cfg/ip/ospf/if <interface number>/aindex <area index>
```

For example, the following commands could be used to configure IP interface 14 for a presence on the 10.10.10.1/24 network, to define OSPF area 1, and to attach the area to the network:

```
>> # /cfg/ip/if 14                          (Select menu for IP interface 14)
>> IP Interface 14# addr 10.10.10.1         (Define IP address on backbone
                                             network)
>> IP Interface 14# enable                  (Enable IP interface 14)
>> IP Interface 14# ../ospf/aindex 1        (Select menu for area index 1)
>> OSPF Area (index) 1 # areaid 0.0.0.1     (Define area ID as OSPF area 1)
>> OSPF Area (index) 1 # enable             (Enable area index 1)
>> OSPF Area (index) 1 # ../if 14           (Select OSPF menu for interface 14)
>> OSPF Interface 14# aindex 1              (Attach area to network on interface
                                             14)
>> OSPF Interface 14# enable                (Enable interface 14 for area index 1)
```

Interface Cost

The OSPF link-state algorithm (Dijkstra's algorithm) places each routing device at the root of a tree and determines the cumulative *cost* required to reach each destination. Usually, the cost is inversely proportional to the bandwidth of the interface. Low cost indicates high bandwidth. You can manually enter the cost for the output route with the following command:

```
>> # /cfg/ip/ospf/if <OSPF interface number>/cost <cost value (1-65535)>
```

Electing the Designated Router and Backup

In any area with more than two routing devices, a Designated Router (DR) is elected as the central contact for database exchanges among neighbors, and a Backup Designated Router (BDR) is elected in case the DR fails.

DR and BDR elections are made through the hello process. The election can be influenced by assigning a priority value to the Web switch's OSPF interfaces. The command is as follows:

```
>> # /cfg/ip/ospf/if <OSPF interface number>/prio <priority value (0-127)>
```

A priority value of 127 is the highest, and 1 is the lowest. A priority value of 0 specifies that the interface cannot be used as a DR or BDR. In case of a tie, the routing device with the lowest router ID wins.

Summarizing Routes

Route summarization condenses routing information. Without summarization, each routing device in an OSPF network would retain a route to every subnet in the network. With summarization, routing devices can reduce some sets of routes to a single advertisement, reducing both the load on the routing device and the perceived complexity of the network. The importance of route summarization increases with network size.

Summary routes can be defined for up to 64 IP address ranges using the following command:

```
>> # /cfg/ip/ospf/range <range number>/addr <IP address>/mask
<mask>
```

where <*range number*> is a number from 1 to 64, <*IP address*> is the base IP address for the range, and <*mask*> is the IP address mask for the range. For a detailed configuration example, see "Example 3: Summarizing Routes" on page 66.

Default Routes

When an OSPF routing device encounters traffic for a destination address it does not recognize, it forwards that traffic along the *default route*. Typically, the default route leads upstream toward the backbone until it reaches the intended area or an external router.

Each Alteon 708 Web switch acting as an ABR automatically inserts a default route into each attached area. In simple OSPF stub areas or NSSAs with only one ABR leading upstream (see Area 1 in Figure 4-3), any traffic for IP address destinations outside the area is forwarded to the switch's IP interface, and then into the connected transit area (usually the backbone). Since this is automatic, no further configuration is required for such areas.

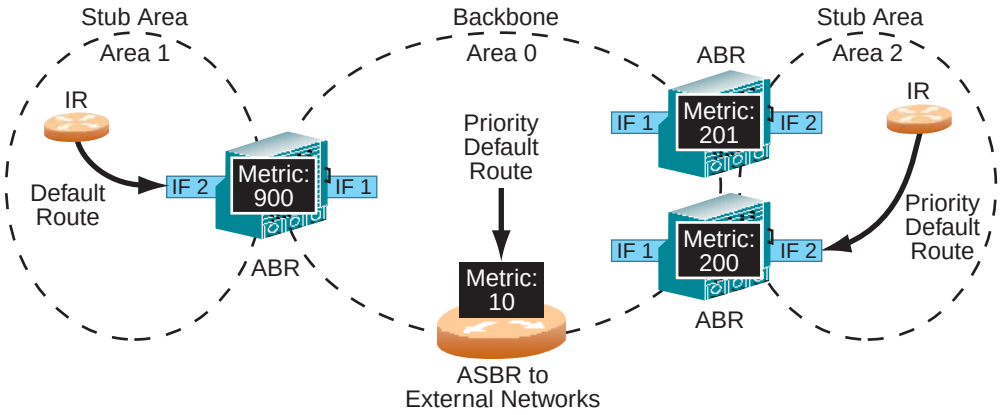


Figure 4-3 Injecting Default Routes

In more complex OSPF areas with multiple ABRs or ASBRs (such as area 0 and area 2 in Figure 4-3), there are multiple routes leading from the area. In such areas, traffic for unrecognized destinations cannot tell which route leads upstream without further configuration.

To select one default route among multiple choices in an area, you can manually configure a metric value on each ABR. The metric assigns a priority to the ABR for its selection as the priority default route in an area. The following command sets the metric value:

```
>> # /cfg/ip/ospf/default <metric value> <metric type (1 or 2)>
```

where <metric value> sets the priority for choosing this switch for default route. The value `none` sets no default and 1 sets the highest priority for default route. Metric type determines the method for influencing routing decisions for external routes.

To clear a default route metric from the switch, use the following command:

```
>> # /cfg/ip/ospf/default none
```

Virtual Links

Usually, all areas in an OSPF AS are physically connected to the backbone. In some cases where this is not possible, you can use a *virtual link*. Virtual links are created to connect one area to the backbone through another non-backbone area (see Figure 4-1 on page 45).

The area which contains a virtual link must be a transit area and have full routing information. Virtual links cannot be configured inside a stub area or NSSA. The area type must be defined as `transit` using the following command:

```
>> # /cfg/ip/ospf/aindex <area index>/type transit
```

The virtual link must be configured on the routing devices at each endpoint of the virtual link, though they may traverse multiple routing devices. To configure an Alteon 708 Web switch as one endpoint of a virtual link, use the following command:

```
>> # /cfg/ip/ospf/virt <link number>/aindex <area index>/nbr <router ID>
```

where <*link number*> is a value between 1 and 8, <*area index*> is the OSPF area index of the transit area, and <*router ID*> is the IP address of the virtual neighbor (nbr), the routing device at the target endpoint. Another router ID is needed when configuring a virtual link in the other direction. To provide the Alteon 708 Web switch with a router ID, see the following section Router ID configuration example.

For a detailed configuration example, see “Example 2: Virtual Links” on page 62.

Router ID

Routing devices in OSPF areas are identified by a router ID. The router ID is expressed in IP address format. The IP address of the router ID is not required to be included in any IP interface range or in any OSPF area.

The router ID can be configured in one of the following two ways:

- Statically—Use the following command to manually configure the router ID:

  ```
  >> # /cfg/ip/rtrid <IP address>
  ```

- Dynamically—OSPF protocol configures the lowest IP interface IP address as the router ID. This is the default. To use a dynamic router ID after having set it statically, set the router ID to 0.0.0.0 and reboot the Web switch.

Authentication

OSPF protocol exchanges are authenticated so that only trusted routing devices can participate. Web OS 7.2 supports only simple authentication (type 1 plain text passwords) among neighboring routing devices within an area.

OSPF passwords are enabled or disabled on the Web switch using the following commands:

```
>> # /cfg/ip/ospf/auth password     (Turn on OSPF password authentication)
>> # /cfg/ip/ospf/auth none         (Turn off OSPF password authentication
```

Plain text passwords up to eight characters long can be configured individually for each OSPF IP interface and virtual link using the following commands:

```
>> # /cfg/ip/ospf/if <OSPF interface number>/key <password>
>> # /cfg/ip/ospf/virt <link number>/key <password>
```

Host Routes for Load Balancing

The Web OS 7.2 implementation of OSPF includes host routes. Host routes are used for advertising network device IP addresses to external networks, accomplishing the following goals:

- Server Load Balancing (SLB) within OSPF

  Host routes advertise virtual server IP addresses to external networks. This allows standard SLB between the Web switch and the server pools in an OSPF environment. For more information on SLB, see Chapter 7, "Server Load Balancing" and your *Web OS 7.2 Command Reference*.

- ABR Load Sharing

  As a second form of load balancing, host routes can be used for dividing OSPF traffic among multiple ABRs. To accomplish this, each Web switch provides identical services but advertises a host route for a different virtual server IP address to the external network. If each virtual server IP address serves a different and equal portion of the external world, incoming traffic from the upstream router should be split evenly among ABRs.

- ABR Failover

  Complementing ABR load sharing, identical host routes can be configured on each ABR. These host routes can be given different costs so that a different ABR is selected as the preferred route for each virtual server and the others are available as backups for failover.

If redundant routes via multiple routing processes (such as OSPF, RIP, BGP, or static routes) exist on your network, the Web switch defaults to the OSPF-derived route.

For a configuration example, see "Example 4: Host Routes" on page 68.

Redistributing Routes into OSPF

Web OS software allows your switch to emulate an ASBR by redistributing information from other routing protocols (static, RIP, iBGP, eBGP, and fixed routes) into OSPF. For information on ASBR, see "Types of OSPF Routing Devices" on page 45. For example, you can instruct OSPF to readvertise a RIP-derived route into OSPF as an AS-External LSA. Based on this LSA, other routers in the OSPF routing domain will install an OSPF route.

Use the following command to redistribute a protocol into OSPF:

`/cfg/ip/ospf/redist <`*`protocol name`***`>`**

where the *protocol name* is static, RIP, iBGP, eBGP, or fixed.

By default, these protocol routes are not redistributed into OSPF.

Use one of the following three methods to redistribute the routes of a particular protocol into OSPF:

- Exporting all the routes of the protocol
- Using route maps

  Route maps allow you to control the redistribution of routes between routing domains. For conceptual information on route maps, see "What is a Route Map?" on page 78.
- Exporting all routes of the protocol except a few selected routes

Each of these methods is discussed in detail in the following sections.

Exporting All Routes

To redistribute all routes of a protocol, use the following command:

`/cfg/ip/ospf/redist <`*`protocol name`***`>/export <`***`metric`***`> <`***`metric type`***`>`**

where metric sets the OSPF cost for the route and *metric type* (either 1 or 2) determines whether the route's cost includes or excludes external costs of the route. If you want to remove a previous configuration to export all the routes of a protocol, then add the parameter `none` to the export command.

`/cfg/ip/ospf/redist <`*`protocol name`***`>/export none`**

Using Route Maps to Export Selected Routes

Use route maps to specify which routes of the protocol you want exported into OSPF. Table 4-1 shows the tasks that you can perform using route maps.

Table 4-1 Commands for Using Route Maps

Task	Command
To add a route map for a particular protocol	**/cfg/ip/ospf/redist <***protocol name***>/addo <***route map numbers***>**
To add all the 32 route maps	**/cfg/ip/ospf/redist <***protocol name***>/addo all**
To remove route maps for a particular protocol	**/cfg/ip/ospf/redist <***protocol name***>/remo <***route map numbers***>**
To remove all 32 route maps for a particular protocol	**/cfg/ip/ospf/redist <***protocol name***>/remo all**

To configure route maps, see "Incoming and Outgoing Route Maps" on page 79 and "Configuration Overview" on page 79.

OSPF does not require you to set all the fields in the route map menu. However, set the following parameters in the route maps and network filter menu:

1. **Enable the route map and assign the metric value in the AS-External LSA.**

```
>> # /cfg/ip/rmap <route map number>/ena
>> # metric <metric value>
```

If a route map is added to a protocol for redistribution, and if the routes of that protocol match any of the routes in the access lists, and if action is set to permit, then those routes are redistributed into OSPF using the metric and metric type assigned for that route map. Metric sets the priority for choosing this switch for default route.

2. **Enable the access list and set the action to `permit`.**

```
>> # alist <access list number>/ena
>> # action permit
```

To redistribute routes matched by the route map, the action in the `alist` must be set to `permit`. If the action is set to `deny` the routes matched by the route map will not be redistributed.

3. **Create and link a network filter to the access list.**

```
>> # nwf <network filter number>
>> # /cfg/ip/nwf <network filter number>/ena
>> # addr <IP address>/mask <IP mask>
```

Optional Parameters for Route Maps

Set the following optional parameters (metric type and metric) for route redistribution into OSPF:

1. **Assign the metric type in the AS-External LSA.**

```
>> # /cfg/ip/rmap <route map number>
>> # type [1|2]
```

Metric type determines the method for influencing routing decisions for external routes.

2. **Match the metric of the protocol route.**

```
>> # alist <access list number>/metric <metric value>
```

M*etric value* sets the priority for choosing this switch for the route. The value `none` sets no default and 1 sets the highest priority for the route.

Exporting All Routes Except a Few Selected Routes

This method is a combination of the previous two methods. The basic steps to configure this method are outlined below:

1. **Configure OSPF to export all routes of the protocol using the export command as described in the first method ("Exporting All Routes" on page 56).**

2. **Use route maps to configure routes to be denied by setting the action in the access list of the route map to `deny`.**

   The configuration of the route map is similar to that described in the second method except that the action is set to `deny`.

OSPF Features Not Supported in This Release

- External route summarization
- Filtering OSPF routes
- Configuring equal cost route load balancing
- Using OSPF to forward multicast routes

OSPF Configuration Examples

A summary of the basic steps for configuring OSPF on the Web switch is listed here. Detailed instructions for each of the steps is covered in the following sections:

1. **Configure IP interfaces.**

   One IP interface is required for each desired network (range of IP addresses) being assigned to an OSPF area on the Web switch.

2. **(Optional) Configure the router ID.**

   The router ID is required only when configuring virtual links on the Web switch.

3. **Enable OSPF on the switch.**
4. **Define the OSPF areas.**
5. **Configure OSPF interface parameters.**

   IP interfaces are used for attaching networks to the various areas.

6. **(Optional) Configure route summarization between OSPF areas.**
7. **(Optional) Configure route summarization when redistributing routes into OSPF.**
8. **(Optional) Configure virtual links.**
9. **(Optional) Configure host routes.**

Example 1: Simple OSPF Domain

In this example, two OSPF areas are defined—one area is the backbone and the other is a stub area. A stub area does not allow advertisements of external routes, thus reducing the size of the database. Instead, a default summary route of IP address 0.0.0.0 is automatically inserted into the stub area. Any traffic for IP address destinations outside the stub area will be forwarded to the stub area's IP interface, and then into the backbone.

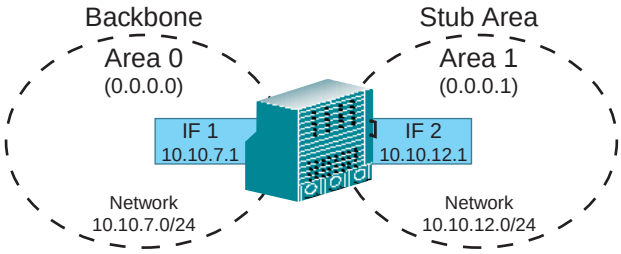


Figure 4-4 A Simple OSPF Domain

Follow this procedure to configure OSPF support as shown in Figure 4-4:

1. **Configure IP interfaces on each network that will be attached to OSPF areas.**

   In this example, two IP interfaces are needed: one for the backbone network on 10.10.7.0/24 and one for the stub area network on 10.10.12.0/24.

```
>> # /cfg/ip/if 1                          (Select menu for IP interface 1)
>> IP Interface 1 # addr 10.10.7.1         (Set IP address on backbone network)
>> IP Interface 1 # mask 255.255.255.0     (Set IP mask on backbone network)
>> IP Interface 1 # broad 10.10.7.255      (Set the broadcast address)
>> IP Interface 1 # enable                 (Enable IP interface 1)
>> IP Interface 1 # ../if 2                (Select menu for IP interface 2)
>> IP Interface 2 # addr 10.10.12.1        (Set IP address on stub area network)
>> IP Interface 2 # enable                 (Enable IP interface 2)
```

2. **Enable OSPF.**

```
>> IP Interface 2 # /cfg/ip/ospf/on        (Enable OSPF on the Web switch)
```

3. **Define the backbone.**

   The backbone is always configured as a transit area using `areaid 0.0.0.0`.

```
>> Open Shortest Path First # aindex 0          (Select menu for area index 0)
>> OSPF Area (index) 0 # areaid 0.0.0.0        (Set the ID for backbone area 0)
>> OSPF Area (index) 0 # type transit          (Define backbone as transit type)
>> OSPF Area (index) 0 # enable                (Enable the area)
```

4. **Define the stub area.**

```
>> OSPF Area (index) 0 # ../aindex 1           (Select menu for area index 1)
>> OSPF Area (index) 1 # areaid 0.0.0.1        (Set the area ID for OSPF area 1)
>> OSPF Area (index) 1 # type stub             (Define area as stub type)
>> OSPF Area (index) 1 # enable                (Enable the area)
```

5. **Attach the network interface to the backbone.**

```
>> OSPF Area 1 # ../if 1                       (Select OSPF menu for IP interface 1)
>> OSPF Interface 1 # aindex 0                 (Attach network to backbone index)
>> OSPF Interface 1 # enable                   (Enable the backbone interface)
```

6. **Attach the network interface to the stub area.**

```
>> OSPF Interface 1 # ../if 2                  (Select OSPF menu for IP interface 2)
>> OSPF Interface 2 # aindex 1                 (Attach network to stub area index)
>> OSPF Interface 2 # enable                   (Enable the stub area interface)
```

7. **Apply and save the configuration changes.**

```
>> OSPF Interface 2 # apply                    (Global command to apply all changes)
>> OSPF Interface 2 # save                     (Global command to save all changes)
```

Example 2: Virtual Links

In the example shown in Figure 4-5, area 2 is not physically connected to the backbone as is usually required. Instead, area 2 will be connected to the backbone via a virtual link through area 1. The virtual link must be configured at each endpoint.

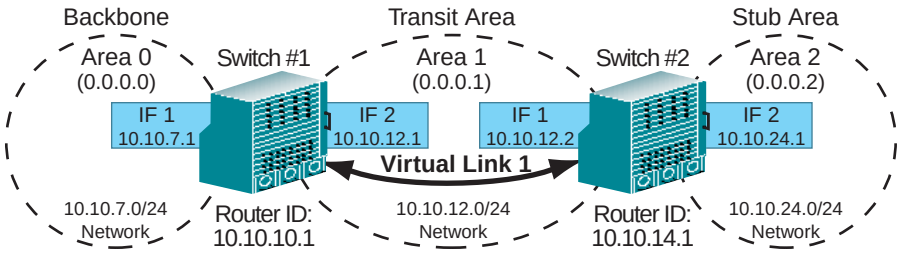


Figure 4-5 Configuring a Virtual Link

Configuring OSPF for a Virtual Link on Switch 1

1. **Configure IP interfaces on each network that will be attached to the switch.**

   In this example, two IP interfaces are needed on Switch 1: one for the backbone network on 10.10.7.0/24 and one for the transit area network on 10.10.12.0/24.

```
>> # /cfg/ip/if 1                    (Select menu for IP interface 1)
>> IP Interface 1 # addr 10.10.7.1   (Set IP address on backbone network)
>> IP Interface 1 # enable           (Enable IP interface 1)
>> IP Interface 1 # ../if 2          (Select menu for IP interface 2)
>> IP Interface 2 # addr 10.10.12.0  (Set IP address on transit area network)
>> IP Interface 2 # enable           (Enable interface 2)
```

2. **Configure the router ID.**

   A router ID is required when configuring virtual links. Later, when configuring the other end of the virtual link on Web Switch 2, the router ID specified here will be used as the target virtual neighbor (`nbr`) address.

```
>> IP Interface 2 # /cfg/ip/rtrid 10.10.10.1  (Set static router ID on Web switch 1)
```

3. **Enable OSPF.**

```
>> IP # /cfg/ip/ospf/on                (Enable OSPF on Web switch 1)
```

4. **Define the backbone.**

```
>> Open Shortest Path First # aindex 0      (Select menu for area index 0)
>> OSPF Area (index) 0 # areaid 0.0.0.0    (Set the area ID for backbone area 0)
>> OSPF Area (index) 0 # type transit       (Define backbone as transit type)
>> OSPF Area (index) 0 # enable             (Enable the area)
```

5. **Define the transit area.**

The area that contains the virtual link must be configured as a transit area.

```
>> OSPF Area (index) 0 # ../aindex 1        (Select menu for area index 1)
>> OSPF Area (index) 1 # areaid 0.0.0.1    (Set the area ID for OSPF area 1)
>> OSPF Area (index) 1 # type transit       (Define area as transit type)
>> OSPF Area (index) 1 # enable             (Enable the area)
```

6. **Attach the network interface to the backbone.**

```
>> OSPF Area (index) 1 # ../if 1            (Select OSPF menu for IP interface 1)
>> OSPF Interface 1 # aindex 0              (Attach network to backbone index)
>> OSPF Interface 1 # enable                (Enable the backbone interface)
```

7. **Attach the network interface to the transit area.**

```
>> OSPF Interface 1 # ../if 2               (Select OSPF menu for IP interface 2)
>> OSPF Interface 2 # aindex 1              (Attach network to transit area index)
>> OSPF Interface 2 # enable                (Enable the transit area interface)
```

8. **Configure the virtual link.**

```
>> OSPF Interface 2 # ../virt 1             (Specify a virtual link number)
>> OSPF Virtual Link 1 # aindex 1           (Specify the transit area for the virtual link)
>> OSPF Virtual Link 1 # nbr 10.10.14.1     (Specify the router ID of the recipient)
>> OSPF Virtual Link 1 # enable             (Enable the virtual link)
```

The `nbr` router ID configured in this step must be the same as the router ID that will be configured for Switch 2 in Step 2 on page 64.

9. **Apply and save the configuration changes.**

```
>> OSPF Interface 2 # apply                 (Global command to apply all changes)
>> OSPF Interface 2 # save                  (Global command to save all changes)
```

Configuring OSPF for a Virtual Link on Switch 2

1. **Configure IP interfaces on each network that will be attached to OSPF areas.**

   Two IP interfaces are needed on Switch 2: one for the transit area network on 10.10.12.0/24 and one for the stub area network on 10.10.24.0/24.

```
>> # /cfg/ip/if 1                          (Select menu for IP interface 1)
>> IP Interface 1 # addr 10.10.12.2        (Set transit area network IP address)
>> IP Interface 1 # enable                 (Enable IP interface 1)
>> IP Interface 1 # ../if 2                (Select menu for IP interface 2)
>> IP Interface 2 # addr 10.10.24.1        (Set IP address on stub area network)
>> IP Interface 2 # enable                 (Enable IP interface 2)
```

2. **Configure the router ID.**

   A router ID is required when configuring virtual links. This router ID should be the same one specified as the target virtual neighbor (`nbr`) on Web Switch 1 in Step 8 on page 63.

```
>> IP Interface 2 # /cfg/ip/rtrid 10.10.14.1    (Set static router ID on Web switch 2)
```

3. **Enable OSPF.**

```
>> IP# /cfg/ip/ospf/on                   (Enable OSPF on Web switch 2)
```

4. **Define the backbone.**

   This version of Web OS 7.2 requires that a backbone index be configured on the non-backbone end of the virtual link as follows:

```
>> Open Shortest Path First # aindex 0     (Select the menu for area index 0)
>> OSPF Area (index) 0 # areaid 0.0.0.0    (Set the area ID for OSPF area 0)
>> OSPF Area (index) 0 # enable            (Enable the area)
```

5. **Define the transit area.**

```
>> OSPF Area (index) 0 # ../aindex 1       (Select menu for area index 1)
>> OSPF Area (index) 1 # areaid 0.0.0.1    (Set the area ID for OSPF area 1)
>> OSPF Area (index) 1 # type transit      (Define area as transit type)
>> OSPF Area (index) 1 # enable            (Enable the area)
```

6. **Define the stub area.**

```
>> OSPF Area (index) 1 # ../aindex 2      (Select the menu for area index 2)
>> OSPF Area (index) 2 # areaid 0.0.0.2  (Set the area ID for OSPF area 2)
>> OSPF Area (index) 2 # type stub       (Define area as stub type)
>> OSPF Area (index) 2 # enable          (Enable the area)
```

7. **Attach the network interface to the backbone.**

```
>> OSPF Area (index) 2 # ../if 1          (Select OSPF menu for IP interface 1)
>> OSPF Interface 1 # aindex 1            (Attach network to transit area index)
>> OSPF Interface 1 # enable              (Enable the transit area interface)
```

8. **Attach the network interface to the transit area.**

```
>> OSPF Interface 1 # ../if 2             (Select OSPF menu for IP interface 2)
>> OSPF Interface 2 # aindex 2            (Attach network to stub area index)
>> OSPF Interface 2 # enable              (Enable the stub area interface)
```

9. **Configure the virtual link.**

The `nbr` router ID configured in this step must be the same as the router ID that was configured for Web Switch 1 in Step 2 on page 62.

```
>> OSPF Interface 2 # ../virt 1           (Specify a virtual link number)
>> OSPF Virtual Link 1 # aindex 1         (Specify the transit area for the virtual link)
>> OSPF Virtual Link 1 # nbr 10.10.10.1   (Specify the router ID of the recipient)
>> OSPF Virtual Link 1 # enable           (Enable the virtual link)
```

10. **Apply and save the configuration changes.**

```
>> OSPF Interface 2 # apply               (Global command to apply all changes)
>> OSPF Interface 2 # save                (Global command to save all changes)
```

Other Virtual Link Options

- You can use redundant paths by configuring multiple virtual links.
- Only the endpoints of the virtual link are configured. The virtual link path may traverse multiple routers in an area as long as there is a routable path between the endpoints.

Example 3: Summarizing Routes

By default, ABRs advertise all the network addresses from one area into another area. Route summarization can be used for consolidating advertised addresses and reducing the perceived complexity of the network.

If the network IP addresses in an area are assigned to a contiguous subnet range, you can configure the ABR to advertise a single summary route that includes all the individual IP addresses within the area.

The following example shows one summary route from area 1 (stub area) injected into area 0 (the backbone). The summary route consists of all IP addresses from 36.128.192.0 through 36.128.254.255.

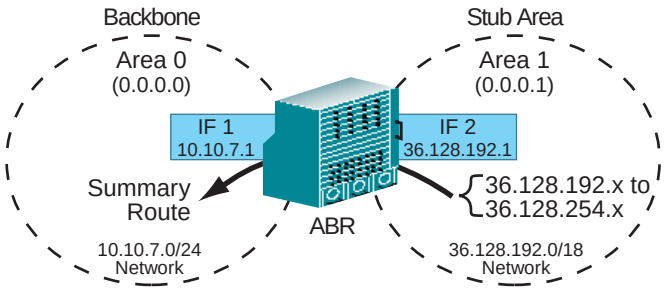


Figure 4-6 Summarizing Routes

NOTE – You can also specify a range of addresses to *prevent* advertising by using the `hide` option.

Follow this procedure to configure OSPF support as shown in Figure 4-6:

1. **Configure IP interfaces for each network which will be attached to OSPF areas.**

```
>> # /cfg/ip/if 1                          (Select menu for IP interface 1)
>> IP Interface 1 # addr 10.10.7.1         (Set IP address on backbone network)
>> IP Interface 1 # enable                 (Enable IP interface 1)
>> IP Interface 1 # ../if 2                (Select menu for IP interface 2)
>> IP Interface 2 # addr 36.128.192.1      (Set IP address on stub area network)
>> IP Interface 2 # enable                 (Enable IP interface 2)
```

2. **Enable OSPF.**

```
>> IP Interface 2 # /cfg/ip/ospf/on        (Enable OSPF on the Web switch)
```

3. **Define the backbone.**

```
>> Open Shortest Path First # aindex 0      (Select menu for area index 0)
>> OSPF Area (index) 0 # areaid 0.0.0.0    (Set the ID for backbone area 0)
>> OSPF Area (index) 0 # type transit      (Define backbone as transit type)
>> OSPF Area (index) 0 # enable            (Enable the area)
```

4. **Define the stub area.**

```
>> OSPF Area (index) 0 # ../aindex 1       (Select menu for area index 1)
>> OSPF Area (index) 1 # areaid 0.0.0.1    (Set the area ID for OSPF area 1)
>> OSPF Area (index) 1 # type stub         (Define area as stub type)
>> OSPF Area (index) 1 # enable            (Enable the area)
```

5. **Attach the network interface to the backbone.**

```
>> OSPF Area (index) 1 # ../if 1           (Select OSPF menu for IP interface 1)
>> OSPF Interface 1 # aindex 0             (Attach network to backbone index)
>> OSPF Interface 1 # enable               (Enable the backbone interface)
```

6. **Attach the network interface to the stub area.**

```
>> OSPF Interface 1 # ../if 2              (Select OSPF menu for IP interface 2)
>> OSPF Interface 2 # aindex 1             (Attach network to stub area index)
>> OSPF Interface 2 # enable               (Enable the stub area interface)
```

7. **Configure route summarization by specifying the starting address and mask of the range of addresses to be summarized.**

```
>> OSPF Interface 2 # ../range 1                  (Select menu for summary range)
>> OSPF Summary Range 1 # addr 36.128.192.0       (Set base IP address of summary range)
>> OSPF Summary Range 1 # mask 255.255.192.0      (Set mask address for summary range)
>> OSPF Summary Range 1 # aindex 0                (Inject summary route into backbone area)
>> OSPF Summary Range 1 # enable                  (Enable summary range)
```

8. **Apply and save the configuration changes.**

```
>> OSPF Summary Range 1 # apply            (Global command to apply all changes)
>> OSPF Summary Range 1 # save             (Global command to save all changes)
```

Example 4: Host Routes

The Web OS 7.2 implementation of OSPF includes host routes. Host routes are used for advertising network device IP addresses to external networks and allows for Server Load Balancing (SLB) within OSPF. It also makes ABR load sharing and failover possible.

Consider the example network in Figure 4-7. Both Web switches have access to servers with identical content and are configured with the same virtual server IP addresses: 10.10.10.1 and 10.10.10.2. Web switch 1 is given a host route with a low cost for virtual server 10.10.10.1 and another host route with a high cost for virtual server 10.10.10.2. Web switch 2 is configured with the same hosts but with the costs reversed; one host route has a high cost for virtual server 10.10.10.1 and another has a low cost for virtual server 10.10.10.2.

All four host routes are injected into the upstream router and advertised externally. Traffic comes in for both virtual server IP addresses (10.10.10.1 and 10.10.10.2). The upstream router sees that both addresses exist on both Web switches and uses the host route with the lowest cost for each. Traffic for 10.10.10.1 goes to Web switch 1 because its host route has the lowest cost for that address. Traffic for 10.10.10.2 goes to Web switch 2 because its host route has the lowest cost. This effectively shares the load among ABRs. Both Web switches then use standard Server Load Balancing (SLB) to distribute traffic among available real servers.

In addition, if one of the Web switches were to fail, the upstream routing device would forward the traffic to the ABR whose host route has the next lowest cost. In this example, the remaining Web switch would assume the entire load for both virtual servers.

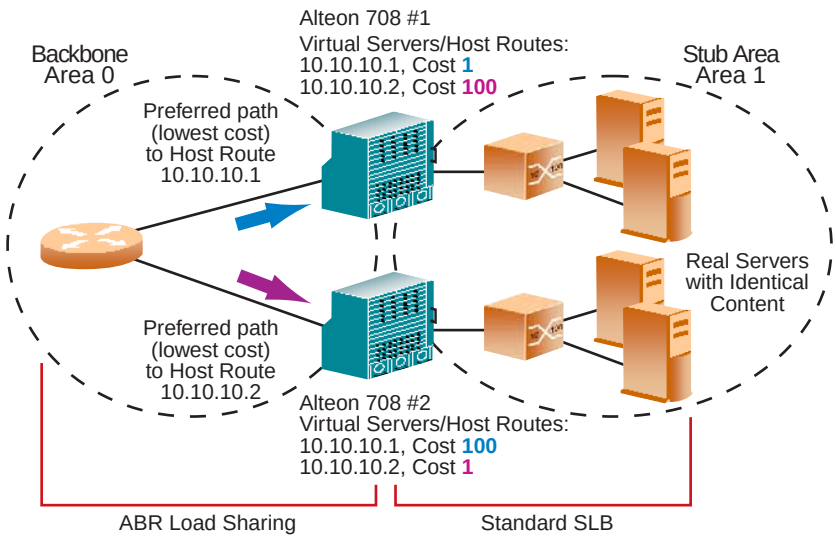


Figure 4-7 Configuring OSPF Host Routes

Configuring OSPF for Host Routes on Alteon 708 Switch 1

1. **Configure basic SLB parameters.**

   Web switch 1 is connected to two real servers. Each real server is given an IP address and is placed in the same real server group.

```
>> # /cfg/slb/real 1                      (Select menu for real server 1)
>> Real server 1 # rip 100.100.100.25     (Set the IP address for real server 1)
>> Real server 1 # enable                 (Enable the real server)
>> Real server 1 # ../real 2              (Select menu for real server 2)
>> Real server 2 # rip 100.100.10.26      (Set the IP address for real server 2)
>> Real server 2 # enable                 (Enable the real server)
>> Real server 2 # ../group 1             (Select menu for real server group 1)
>> Real server group 1 # add 1            (Add real server 1 to group)
>> Real server group 1 # add 2            (Add real server 2 to group)
>> Real server group 1 # enable           (Enable the group)
>> Real server group 1 # ../on            (Turn SLB on)
```

2. **Configure the primary virtual server.**

   Alteon 708 Switch 1 will be preferred for virtual server 10.10.10.1.

```
>> Layer 4 # virt 1                       (Select menu for virtual server 1)
>> Virtual server 1 # vip 10.10.10.1      (Set the IP address for virtual server 1)
>> Virtual server 1 # enable              (Enable the virtual server)
>> Virtual server 1 # service http        (Select menu for service on virtual server)
>> Virtual server 1 http service # group 1 (Use real server group 1 for http service)
```

3. **Configure the backup virtual server.**

   Alteon 708 Switch 1 will act as a backup for virtual server 10.10.10.2. Both virtual servers in this example are configured with the same real server group and provide identical services.

```
>> Virtual server 2 http service # /cfg/slb/virt 2  (Select menu for virtual server 2)
>> Virtual server 1 # vip 10.10.10.2      (Set the IP address for virtual server 2)
>> Virtual server 1 # enable              (Enable the virtual server)
>> Virtual server 1 # service http        (Select menu for service on virtual server)
>> Virtual server 1 # group 1             (Use real server group 1 for http service)
```

4. **Configure IP interfaces for each network that will be attached to OSPF areas.**

```
>> Virtual server 1 # /cfg/ip/if 1          (Select menu for IP interface 1)
>> IP Interface 1 # addr 10.10.7.1          (Set IP address on backbone network)
>> IP Interface 1 # enable                  (Enable IP interface 1)
>> IP Interface 1 # ../if 2                 (Select menu for IP interface 2)
>> IP Interface 2 # addr 10.10.10.40        (Set IP address on stub area network)
>> IP Interface 2 # enable                  (Enable IP interface 2)
```

5. **Enable OSPF on Web switch 1.**

```
>> IP Interface 2 # ../ospf/on              (Enable OSPF on Web switch 1)
```

6. **Define the backbone.**

```
>> Open Shortest Path First # aindex 0      (Select menu for area index 0)
>> OSPF Area (index) 0 # areaid 0.0.0.0     (Set the ID for backbone area 0)
>> OSPF Area (index) 0 # type transit       (Define backbone as transit type)
>> OSPF Area (index) 0 # enable             (Enable the area)
```

7. **Define the stub area.**

```
>> OSPF Area (index) 0 # ../aindex 1        (Select menu for area index 1)
>> OSPF Area (index) 1 # areaid 0.0.0.1     (Set the ID for stub area 1)
>> OSPF Area (index) 1 # type stub          (Define area as stub type)
>> OSPF Area (index) 1 # enable             (Enable the area)
```

8. **Attach the network interface to the backbone.**

```
>> OSPF Area (index) 1 # ../if 1            (Select OSPF menu for IP interface 1)
>> OSPF Interface 1 # aindex 0              (Attach network to backbone index)
>> OSPF Interface 1 # enable                (Enable the backbone interface)
```

9. **Attach the network interface to the stub area.**

```
>> OSPF Interface 1 # ../if 2               (Select OSPF menu for IP interface 2)
>> OSPF Interface 2 # aindex 1              (Attach network to stub area index)
>> OSPF Interface 2 # enable                (Enable the stub area interface)
```

10. Configure host routes.

One host route is needed for each virtual server on Web Switch 1. Since virtual server 10.10.10.1 is preferred for Web switch 1, its host route has a low cost. Because virtual server 10.10.10.2 is used as a backup in case Web switch 2 fails, its host route has a high cost.

```
>> OSPF Interface 2 # ../host 1                 (Select menu for host route 1)
>> OSPF Host Entry 1 # addr 10.10.10.1          (Set IP address same as virtual server 1)
>> OSPF Host Entry 1 # aindex 0                 (Inject host route into backbone area)
>> OSPF Host Entry 1 # cost 1                   (Set low cost for preferred path)
>> OSPF Host Entry 1 # enable                   (Enable the host route)
>> OSPF Host Entry 1 # ../host 2                (Select menu for host route 2)
>> OSPF Host Entry 2 # addr 10.10.10.2          (Set IP address same as virtual server 2)
>> OSPF Host Entry 2 # aindex 0                 (Inject host route into backbone area)
>> OSPF Host Entry 2 # cost 100                 (Set high cost for use as backup path)
>> OSPF Host Entry 2 # enable                   (Enable the host route)
```

11. Apply and save the configuration changes.

```
>> OSPF Host Entry 2 # apply                    (Global command to apply all changes)
>> OSPF Host Entry 2 # save                     (Global command to save all changes)
```

Configuring OSPF for Host Routes on Alteon 708 Switch 2

1. Configure basic SLB parameters.

Web Switch 2 is connected to two real servers. Each real server is given an IP address and is placed in the same real server group.

```
>> # /cfg/slb/real 1                           (Select menu for real server 1)
>> Real server 1 # rip 100.100.100.27          (Set the IP address for real server 1)
>> Real server 1 # enable                      (Enable the real server)
>> Real server 1 # ../real 2                   (Select menu for real server 2)
>> Real server 2 # rip 100.100.10.28           (Set the IP address for real server 2)
>> Real server 2 # enable                      (Enable the real server)
>> Real server 2 # ../group 1                  (Select menu for real server group 1)
>> Real server group 1 # add 1                 (Add real server 1 to group)
>> Real server group 1 # add 2                 (Add real server 2 to group)
>> Real server group 1 # enable                (Enable the group)
>> Real server group 1 # ../on                 (Turn SLB on)
```

2. **Configure the virtual server parameters.**

   The same virtual servers are configured as on Web switch 1.

```
>> Layer 4 # virt 1                              (Select menu for virtual server 1)
>> Virtual server 1 # vip 10.10.10.1             (Set the IP address for virtual server 1)
>> Virtual server 1 # enable                     (Enable the virtual server)
>> Virtual server 1 # service http               (Select menu for service on virtual server)
>> Virtual server 1 http service # group 1       (Use real server group 1 for http service)
>> Virtual server 2 http service # /cfg/slb/virt 2 (Select menu for virtual server 2)
>> Virtual server 1 # vip 10.10.10.2             (Set the IP address for virtual server 2)
>> Virtual server 1 # enable                     (Enable the virtual server)
>> Virtual server 1 # service http               (Select menu for service on virtual server)
>> Virtual server 1 # group 1                    (Use real server group 1 for http service)
```

3. **Configure IP interfaces for each network that will be attached to OSPF areas.**

```
>> Virtual server 1 # /cfg/ip/if 1               (Select menu for IP Interface 1)
>> IP Interface 1 # addr 10.10.7.2               (Set IP address on backbone network)
>> IP Interface 1 # enable                       (Enable IP interface 1)
>> IP Interface 1 # ../if 2                      (Select menu for IP Interface 2)
>> IP Interface 2 # addr 10.10.10.41             (Set IP address on stub area network)
>> IP Interface 2 # enable                       (Enable IP interface 2)
```

4. **Enable OSPF on Web Switch 2.**

```
>> IP Interface 2 # ../ospf/on                   (Enable OSPF on Web switch #2)
```

5. **Define the backbone.**

```
>> Open Shortest Path First # aindex 0           (Select menu for area index 0)
>> OSPF Area (index) 0 # areaid 0.0.0.0         (Set the ID for backbone area 0)
>> OSPF Area (index) 0 # type transit            (Define backbone as transit type)
>> OSPF Area (index) 0 # enable                  (Enable the area)
```

6. **Define the stub area.**

```
>> OSPF Area (index) 0 # ../aindex 1             (Select menu for area index 1)
>> OSPF Area (index) 1 # areaid 0.0.0.1         (Set the ID for stub area 1)
>> OSPF Area (index) 1 # type stub               (Define area as stub type)
>> OSPF Area (index) 1 # enable                  (Enable the area)
```

7. **Attach the network interface to the backbone.**

```
>> OSPF Area (index) 1 # ../if 1          (Select OSPF menu for IP interface 1)
>> OSPF Interface 1 # aindex 0            (Attach network to backbone index)
>> OSPF Interface 1 # enable              (Enable the backbone interface)
```

8. **Attach the network interface to the stub area.**

```
>> OSPF Interface 1 # ../if 2             (Select OSPF menu for IP interface 2)
>> OSPF Interface 2 # aindex 1            (Attach network to stub area index)
>> OSPF Interface 2 # enable              (Enable the stub area interface)
```

9. **Configure host routes.**

Host routes are configured just like those on Web switch 1, except their costs are *reversed*. Since virtual server 10.10.10.2 is preferred for Web switch 2, its host route has been given a low cost. Because virtual server 10.10.10.1 is used as a backup in case Web switch 1 fails, its host route has been given a high cost.

```
>> OSPF Interface 2 # ../host 1           (Select menu for host route 1)
>> OSPF Host Entry 1 # addr 10.10.10.1    (Set IP address same as virtual server 1)
>> OSPF Host Entry 1 # aindex 0           (Inject host route into backbone area)
>> OSPF Host Entry 1 # cost 100           (Set high cost for use as backup path)
>> OSPF Host Entry 1 # enable             (Enable the host route)
>> OSPF Host Entry 1 # ../host 2          (Select menu for host route 2)
>> OSPF Host Entry 2 # addr 10.10.10.2    (Set IP address same as virtual server 2)
>> OSPF Host Entry 2 # aindex 0           (Inject host route into backbone area)
>> OSPF Host Entry 2 # cost 1             (Set low cost for primary path)
>> OSPF Host Entry 2 # enable             (Enable the host route)
```

10. **Apply and save the configuration changes.**

```
>> OSPF Host Entry 2 # apply              (Global command to apply all changes)
>> OSPF Host Entry 2 # save               (Global command to save all changes)
```

Verifying OSPF Support

Use the following commands to verify the OSPF configuration on your switch:

- **/info/ospf/general**
- **/info/ospf/route**
- **/info/ospf/nbr**

CHAPTER 5

Border Gateway Protocol

Border Gateway Protocol (BGP) is an Internet protocol that enables routers on a network to share and advertise routing information about the segments of the IP address space they can access both within their network and with routers on external networks. BGP allows you to decide what is the “best” route for a packet to take from your network to a destination on another network rather than simply using a default route from your border router(s) to your upstream provider(s). BGP is defined in RFC 1771.

The following topics are discussed in this section:

- “Internal Routing Versus External Routing” on page 76
- “Forming BGP Peer Routers” on page 77
- “What is a Route Map?” on page 78
- “Aggregating Routes” on page 81
- “Redistributing Routes” on page 82
- “BGP Attributes” on page 83
- “Selecting Route Paths in BGP” on page 84
- “Configuration Examples” on page 85
- “Web OS BGP Limitations” on page 89

Alteon Web switches can advertise their IP interfaces and virtual IP addresses using BGP and take BGP feeds from as many as 16 BGP peer routers. This allows more resilience and flexibility in balancing traffic from the Internet.

Internal Routing Versus External Routing

To ensure effective processing of network traffic, every router on your network needs to know how to send a packet (directly or indirectly) to any other location/destination in your network. This is referred to as *internal routing* and can be done with static routes or using active, internal routing protocols, such as RIP, RIPv2, and OSPF.

It is also useful to tell routers outside your network (upstream providers or *peers*) about the routes you can access in your network. External networks (those outside your own) that are under the same administrative control are referred to as *Autonomous Systems* (ASs). Sharing of routing information between autonomous systems is known as *external routing*.

External BGP (eBGP) is used to exchange routes between different autonomous systems, whereas internal BGP (iBGP) is used to exchange routes within the same autonomous system. iBGP is a type of internal routing protocol and, like OSPF, you use it to do active routing inside your network. However, iBGP is slower than OSPF in convergency, but it carries AS path information, which is important when you are an ISP or doing BGP transit.

NOTE – The iBGP peers have to be part of a fully meshed network, as shown in Figure 5-1.

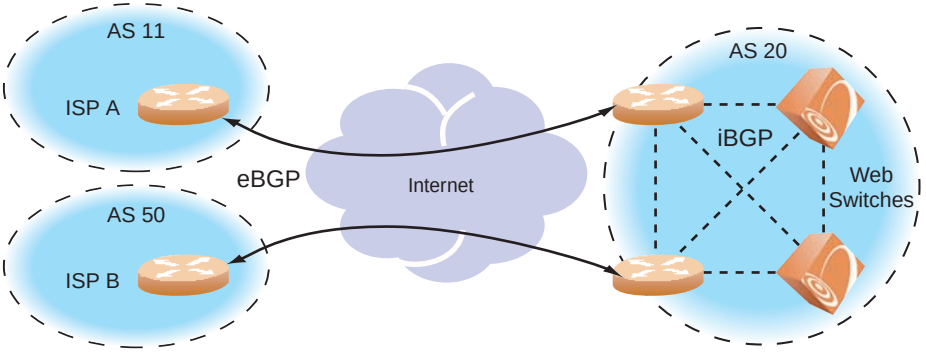


Figure 5-1 iBGP and eBGP

Typically, an AS has one or multiple border routers that exchange routes with other ASs. It also has an internal routing scheme that enables every router in that AS to get to every other router and destination within that AS. Typically, iBGP allows you to control the exit point from an AS.

When you *advertise* routes to border routers on other autonomous systems, you are effectively responsible for carrying data to the IP space represented in the route being advertised. For example, if you advertise 192.204.4.0/24, you are declaring that if another router sends you data destined for any address in 192.204.4.0/24, you know how to carry that data to its destination.

Forming BGP Peer Routers

Two BGP routers become peers or neighbors once you establish a TCP connection between them. If a peer is interested in a route (for example, if a peer would like to receive your new static route and the new route is static), an update message is sent to that peer with the new route. If the previously advertised route is no longer available for use, an update message indicating that the route has been withdrawn is sent to the peer.

The following example (refer to Figure 5-3 on page 85) illustrates the commands used to specify two peer routers:

NOTE – Before configuring peers, specify the local AS and router ID with the **`/cfg/ip/as`** and **`/cfg/ip/rtrid`** commands.

```
>> # /cfg/ip/bgp/peer 1                    (Select BGP peer router 1)
>> BGP Peer 1# ena                         (Enable this peer configuration)
>> BGP Peer 1# addr 200.200.200.2          (Set IP address for peer router 1)
>> BGP Peer 1# ras 100                     (Set remote AS number)
>> BGP Peer 1# ../peer 2                   (Select BGP peer router 2)
>> BGP Peer 2# ena                         (Enable this peer configuration)
>> BGP Peer 2# addr 210.210.210.2          (Set IP address for peer router 2)
>> BGP Peer 2# ras 200                     (Set remote AS number)
```

For each Internet host, you must be able to send a packet out a path to that host, and that host must have a path back to you. Further, whoever provides Internet connectivity to that host must have a path to you. Ultimately, this means that they must “hear a route” that covers the section of the IP space you are using, or you will not have connectivity to the host in question.

What is a Route Map?

A route map is used to control and modify routing information. Route maps define conditions for redistributing routes from one routing protocol to another or controlling routing information when injecting it in and out of BGP. Route maps are used by OSPF only for redistributing routes. For example, a route map is used to set a preference value for a specific route from a peer router and another preference value for all other routes learned via the same peer router. For example, the following command is used to define a route map:

```
>> # /cfg/ip/rmap 1                    (Select a route map)
```

A route map allows you to match attributes, such as metric, network address, and AS number. It also allows users to overwrite the local preference metric and to append the AS number in the AS route. See "BGP Failover Example" on page 85.

Web OS allows you to configure 32 route maps. Each route map can have up to eight access lists. Each access list consists of a network filter. A network filter defines an IP address and subnet mask of the network that you want to include in the filter. Figure 5-2 illustrates the relationship between route maps, access lists and network filters.

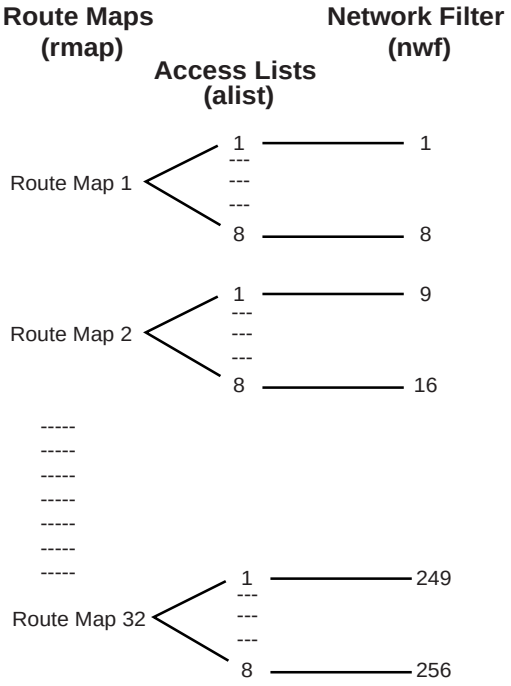


Figure 5-2 Distributing Network Filters in Access Lists and Route Maps

Incoming and Outgoing Route Maps

You can have two types of route maps: incoming and outgoing. A BGP peer router can be configured to support up to eight route maps in the incoming route map list and outgoing route map list.

If a route map is not configured in the incoming route map list, the router imports all BGP updates. If a route map is configured in the incoming route map list, the router ignores all unmatched incoming updates.

Route maps in an outgoing route map list behave similar to route maps in an incoming route map list. If a route map is not configured in the outgoing route map list, all routes are advertised or permitted. If a route map is configured in the outgoing route map list, matched routes are advertised and unmatched routes are ignored.

Precedence

You can set a priority to a route map by specifying a precedence value with the following command:

```
>> /cfg/ip/rmap/pre                    (Specify a precedence)
```

The smaller the value the higher the precedence. If two route maps have the same precedence value, the smaller number has higher precedence.

Configuration Overview

To configure route maps, you need to do the following:

1. **Define network filter.**

```
>> # /cfg/ip/nwf 1                                  (Specify a network filter number)
>> IP Network Filter 1# addr <IP address>          (Specify network address)
>> IP Network Filter 1# mask <IP mask>             (Specify network mask)
>> IP Network Filter 1# ena                        (Enable network filter)
```

Enter a filter number from 1 to 256. Specify the IP address and subnet mask of the network that you want to match. Enable the network filter. You can distribute up to 256 network filters among 32 route maps each containing 8 access lists.

2. **(Optional) Define the criteria for the access list and enable it .**

   Specify the access list and associate the network filter number configured in Step 1.

```
>> # /cfg/ip/rmap 1                    (Specify a route map number)
>> IP Route Map 1# alist 1             (Specify the access list number)
>> IP Access List 1# nwf 1             (Specify the network filter number)
>> IP Access List 1# metric            (Define a metric)
>> IP Access List 1# action deny       (Specify action for the access list)
>> IP Access List 1# ena               (Enable the access list)
```

   Steps 2 and 3 are optional, depending on the criteria that you want to match. In Step 2, the network filter number is used to match the subnets defined in the network filter. In Step 3, the autonomous system number is used to match the subnets. Or, you can use both (Step 2 and Step 3) criteria: access list (network filter) and access path (AS filter) to configure the route maps.

3. **(Optional) Configure the attributes in the AS filter menu.**

```
>> # cfg/ip/rmap 1/aspath               (Specify the attributes in the filter)
```

   Define the AS filter number and the attributes for the filter. Enable the filter.

4. **Set up the BGP attributes.**

   If you want to overwrite the attributes that the peer router is sending, then define the following BGP attributes:

   - Specify the AS numbers that you want to prepend to a matched route and the local preference for the matched route.
   - Specify the metric [Multi Exit Discriminator (MED)] for the matched route.

```
>> # cfg/ip/rmap 1                     (Specify a route map number)
>> IP Route Map 1# ap                  (Specify the AS numbers to prepend)
>> IP Route Map 1# lp                  (Specify the local preference)
>> IP Route Map 1# med                 (Specify the metric)
```

5. **Enable the route map.**

```
>> # cfg/ip/rmap 1/en                  (Enable the route map)
```

6. **Assign the route map to a peer router.**

   Select the peer router and then add the route map to the incoming route map list,

```
>> # cfg/ip/bgp/peer 1/addi          (Add to the incoming route map)
```

   or to the outgoing route map list.

```
>> # cfg/ip/bgp/peer 1/addo          (Add to the outgoing route map)
```

Aggregating Routes

Aggregation is the process of combining several different routes in such a way that a single route can be advertised, which minimizes the size of the routing table. You can configure aggregate routes in BGP either by redistributing an aggregate route into BGP or by creating an aggregate entry in the BGP routing table.

When a subnet is redistributed from an Interior Gateway Protocol (IGP) into BGP, only the network route is injected into the BGP table. By default, this automatic summarization is disabled. To define the route to aggregate, use the following commands:

```
>> # cfg/ip/bgp                          (Specify BGP)
>> Border Gateway Protocol# aggr 1       (Specify aggregate list number)
>> BGP aggr 1 # addr                     (Enter aggregation network address)
>> BGP aggr 1 # mask                     (Enter aggregation network mask)
>> BGP aggr 1 # ena                      (Enable aggregation)
```

An example of creating a BGP aggregate route is shown in "Default Redistribution and Route Aggregation Example" on page 88.

Redistributing Routes

In addition to running multiple routing protocols simultaneously, Web OS software can redistribute information from one routing protocol to another. For example, you can instruct the switch to use OSPF to readvertise RIP-derived routes or static routes. This applies to all of the IP-based routing protocols.

You can also conditionally control the redistribution of routes between routing domains by defining a method known as route maps between the two domains. For more information on route maps, see "What is a Route Map?" on page 78. Redistributing routes is another way of providing policy control over whether to export RIP routes, OSPF routes, fixed routes, static routes, and virtual IP address routes. Default routes can be configured using the following methods:

- Import
- Originate

  The router sends a default route to peers even though it does not have any default routes in its routing table.

- Redistribute

  Default routes are either configured through the default gateway or learned via other protocols and redistributed to peer routers. If the default routes are from the default gateway, enable the static routes because default routes from the default gateway are static routes. Similarly, if the routes are learned from another routing protocol, make sure you enable that protocol for redistribution.

- None

BGP Attributes

The following two BGP attributes are discussed in this section: Local preference and metric (Multi-Exit Discriminator).

Local Preference Attribute

When there are multiple paths to the same destination, the local preference attribute indicates the preferred path. The path with the higher preference is preferred (the default value of the local preference attribute is 100). Unlike the weight attribute, which is only relevant to the local router, the local preference attribute is part of the routing update and is exchanged among routers in the same AS.

The local preference attribute can be set in one of two ways:

- **`/cfg/ip/bgp/pref`**

  This command uses the BGP default local preference method, affecting the outbound direction only.

- **`/cfg/ip/rmap/lp`**

  This command uses the route map local preference method, which affects both inbound and outbound directions.

Metric (Multi-Exit Discriminator) Attribute

This attribute is a hint to external neighbors about the preferred path into an AS when there are multiple entry points. A lower metric value is preferred over a higher metric value. The default value of the metric attribute is 0.

Unlike local preference, the metric attribute is exchanged between ASs; however, a metric attribute that comes into an AS does not leave the AS.

When an update enters the AS with a certain metric value, that value is used for decision making within the AS. When BGP sends that update to another AS, the metric is reset to 0.

Unless otherwise specified, the router compares metric attributes for paths from external neighbors that are in the same AS.

Selecting Route Paths in BGP

BGP selects only one path as the best path. It does not rely on metrics attributes to determine the best path. When the same network is learned via more than one BGP peer, BGP uses its policy for selecting the best route to that network. The BGP implementation on the Alteon 708 switches uses the following criteria to select a path when the same route is received from multiple peers.

1. **Local fixed and static routes are preferred over learned routes.**
2. **With iBGP peers, routes with higher local preference values are selected.**
3. **In the case of multiple routes of equal preference, the route with lower AS path weight is selected.**

   AS path weight = 128 x AS path length (number of autonomous systems transversed).
4. **In the case of equal weight and routes learned from peers that reside in the same AS, the lower metric is selected.**

NOTE – A route with a metric is preferred over a route without a metric.

---

5. **The lower cost to the next hop of routes is selected.**
6. **In the case of equal cost, the eBGP route is preferred over iBGP.**
7. **If all routes are from eBGP, the route with the lower router ID is selected.**

   When the path is selected, BGP puts the selected path in its routing table and propagates the path to its neighbors.

Configuration Examples

This section provides a step-by-step procedure to configure two examples:

- BGP Failover
- Default route redistribution and route aggregation

BGP Failover Example

The following example shows you how to create redundant default gateways for an Alteon 708 at a Web Host/ISP site so that if one gateway should go down, requests will still be forwarded to an upstream router unknown to the switch.

As shown in Figure 5-3, one Alteon 708 is connected to two default gateways: ISP 1 and ISP 2. The customer negotiates with two ISPs to allow the Alteon 708 to use their peer routers as default gateways. The ISP peer routers will then need to announce themselves as default gateways to the switch.

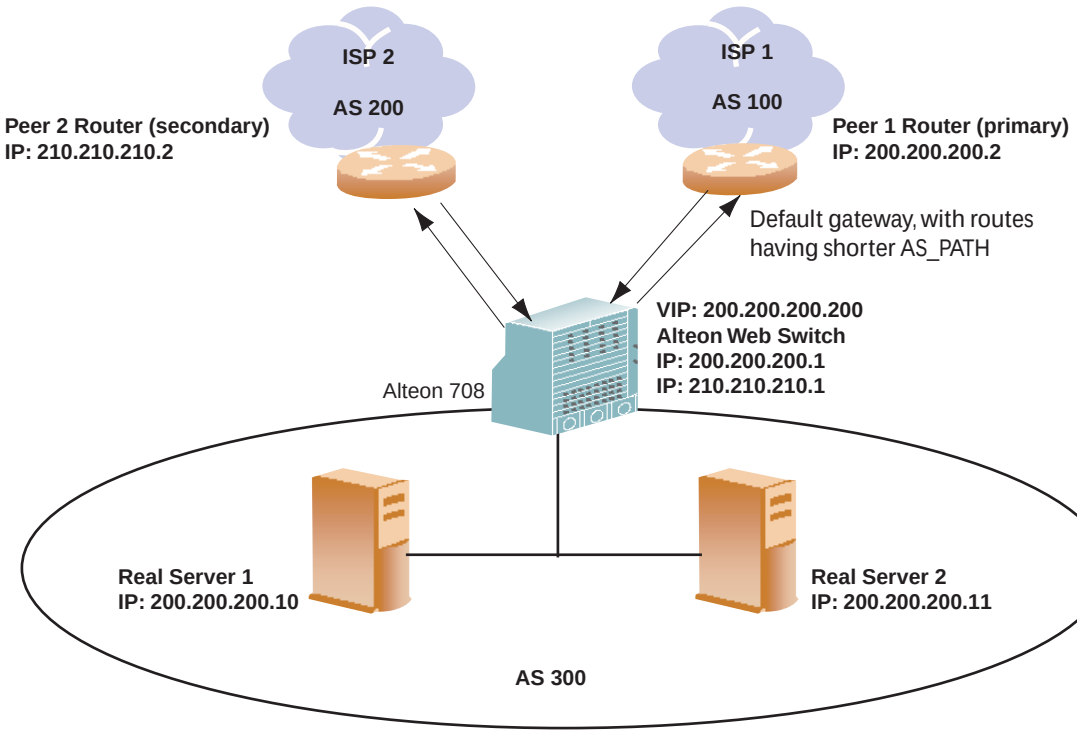


Figure 5-3 BGP Failover Configuration Example

On the switch, one peer router (the secondary) is configured with a longer AS path than the other, so that the peer with the shorter AS path will be seen by the switch as the primary default gateway. ISP 2, the secondary peer, is configured in a route map, thereby appearing to the switch as three router *hops* away.

1. **Configure the switch as you normally would for Server Load Balancing (SLB).**

   - Assign an IP address to each of the real servers in the server pool.
   - Define each real server.
   - Define a real server group.
   - Define a virtual server.
   - Define the port configuration.

   For more information about SLB configuration, refer to Chapter 7, “Server Load Balancing".

2. **Define the VLANs.**

   For simplicity, both default gateways are configured in the same VLAN in this example. The gateways could be in the same VLAN or different VLANs.

```
>> # /cfg/vlan 1                          (Select VLAN 1)
>> vlan 1# add <port number>              (Add a port to the VLAN membership)
>> vlan 1# ena                            (Enable VLAN 1)
```

3. **Define the IP interfaces.**

   The switch will need an IP interface for each default gateway to which it will be connected. Each interface will need to be placed in the appropriate VLAN. These interfaces will be used as the primary and secondary default gateways for the switch.

```
>> # /cfg/ip/rearp 10                     (Set re-ARP period for interface to 10)
>> IP# metrc strict                       (Set metric for default gateway)
>> IP# if 1                               (Select default gateway interface 1)
>> IP Interface 1# ena                    (Enable switch interface 1)
>> IP Interface 1# addr 200.200.200.1     (Configure IP address of interface 1)
>> IP Interface 1# broad 200.200.200.255  (Configure IP broadcast address)
>> IP Interface 1# vlan 1                 (Configure VLAN # for this interface)
>> IP Interface 1# ../if 2                (Select default gateway interface 2)
>> IP Interface 2# ena                    (Enable switch interface 2)
>> IP Interface 2# addr 210.210.210.1     (Configure IP address of interface 2)
>> IP Interface 2# broad 210.210.210.255  (Configure IP broadcast address)
>> IP Interface 2# vlan 1                 (Configure VLAN # for this interface)
```

To help eliminate the possibility for a Denial of Service (DoS) attack, the forwarding of directed broadcasts is disabled by default.

4. **Configure BGP Peer router 1 and 2.**

   Peer 1 is the primary gateway router. Peer 2 is configured in the incoming route map list. The `metric` option is key to ensuring gateway traffic is directed to Peer 1, as it will make Peer 2 appear to be three router hops away from the switch. Thus, the switch should never use Peer 2 unless Peer 1 goes down.

```
>> # /cfg/ip/bgp/peer 1                    (Select BGP peer router 1)
>> BGP Peer 1# ena                         (Enable this peer configuration)
>> BGP Peer 1# addr 200.200.200.2          (Set IP address for peer router 1)
>> BGP Peer 1# ras 100                     (Set remote AS number)
>> BGP Peer 1# redist/default import       (Import default routes)
>> BGP Peer 1# ../peer 2                   (Select BGP peer router 2)
>> BGP Peer 2# ena                         (Enable this peer configuration)
>> BGP Peer 2# addr 210.210.210.2          (Set IP address for peer router 2)
>> BGP Peer 2# ras 200                     (Set remote AS number)
>> BGP Peer 2# redist/default import       (Import default routes)
```

5. **Configure route map for Peer 2.**

   The route map is configured based on the AS number. After you specify the AS path number, repeat the AS number three times to make it three more hops away.

```
>> # /cfg/ip/rmap 1                        (Configure route map 1)
>> IP Route Map 1# aspath 1                (Configure the AS filter number)
>> AS Filter 1# as 200                     (Set the AS number)
>> AS Filter 1# ena                        (Enable this AS filter)
>> IP Route Map 1# ap 200 200 200          (Prepend the AS number to the path)
>> IP Route Map 1# ena                     (Enable the route map)
```

6. **Add configured route map to incoming route map list.**

```
>> /cfg/ip/bgp/peer 2                      (Select the BGP peer router 2)
>> BGP Peer 2# addi 1                      (Add to incoming route map list)
```

7. **On the switch, apply and save your configuration changes.**

```
>> BGP Peer 2# apply                       (Make your changes active)
>> save                                    (Save for restore after reboot)
```

Default Redistribution and Route Aggregation Example

This example shows you how to configure the switch to redistribute information from one routing protocol to another and create an aggregate route entry in the BGP routing table to minimize the size of the routing table.

As illustrated in Figure 5-4, you have two peer routers: an internal and an external peer router. Configure the Alteon 708 switch to redistribute the default routes from AS 200 to AS 135. At the same time, configure for route aggregation to allow you to condense the number of routes traversing from AS 135 to AS 200.

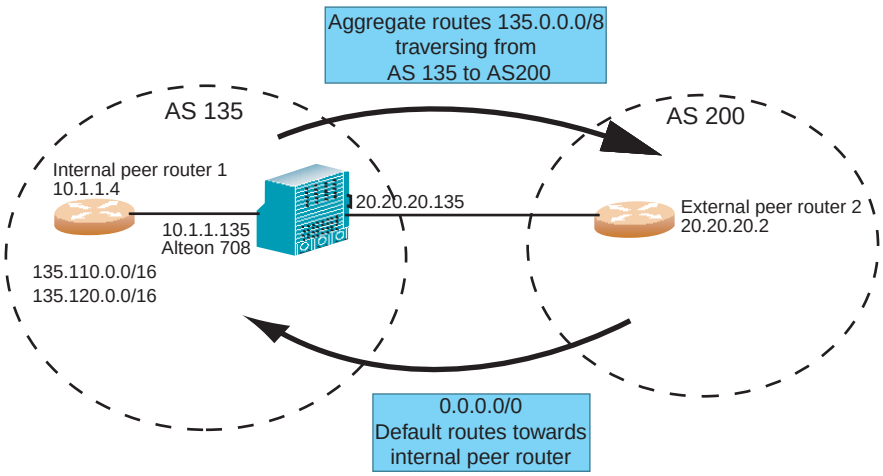


Figure 5-4 Route Aggregation and Default Route Redistribution

1. **Configure the IP interface.**

2. **Configure the local AS number (AS 135) and router ID number (10.1.1.135).**

```
>> # /cfg/ip                    (Select IP)
>> IP# as 135                  (Specify an AS number)
>> IP# rtrid                   (Specify a router ID number)
```

3. **Configure internal peer router 1 and external peer router 2.**

```
>> # /cfg/ip/bgp/peer 1                    (Select internal peer router 1)
>> BGP Peer 1# ena                         (Enable this peer configuration)
>> BGP Peer 1# addr 10.1.1.4               (Set IP address for peer router 1)
>> BGP Peer 1# ras 135                     (Set remote AS number)
>> BGP Peer 1# ../peer 2                   (Select external peer router 2)
>> BGP Peer 2# ena                         (Enable this peer configuration)
>> BGP Peer 2# addr 20.20.20.2             (Set IP address for peer router 2)
>> BGP Peer 2# ras 200                     (Set remote AS number)
```

4. **Configure redistribution for Peer 1.**

```
>> ../bgp peer 1/redist                    (Select redistribute)
>> BGP Peer 1# default redistribute        (Set default to redistribute)
>> BGP Peer 1# fixed ena                   (Enable fixed routes)
```

5. **Configure aggregation policy control.**

Configure the routes that you want aggregated.

```
>> ../aggr 1                               (Set aggregation number)
>> BGP aggr 1# addr 135.0.0.0              (Add IP address to aggregate 1)
>> BGP Peer 1# mask 255.0.0.0              (Add IP mask to aggregate 1)
```

Web OS BGP Limitations

BGP implemented in Web OS supports standard base attributes, internal (iBGP) and external (eBGP) peers, as described in RFC 1771. However, BGP extensions—such as Route Reflection, AS Confederation, Route Flap Dampening, Communities, MD-5 Authentication, and Multi-protocol—are not supported in this release.

CHAPTER 6

Secure Switch Management

To limit access to the switch's Management Processor without having to configure filters for each switch port, you can set a source IP address (or range) that will be allowed to connect to the switch IP interface through Telnet, SSH, SNMP, or the Web OS Browser-Based Interface (BBI). This will also help prevent spoofing or attacks on the switch's TCP/IP stack. The following sections are addressed in this chapter:

- "Setting Allowable Source IP Address Range" on page 92
- "Secure Switch Management" on page 93
- "RADIUS Authentication and Authorization" on page 94
- "Secure Shell and Secure Copy" on page 99

Setting Allowable Source IP Address Range

The allowable management IP address range is configured using the system `mnet` and `mmask` options available on the Command Line Interface (CLI) system menu (`/cfg/sys`).

NOTE – The `mnet` and `mmask` commands in the `/cfg/slb` menu are used for a different purpose.

When an IP packet reaches the Management Processor, the source IP address is checked against the range of addresses defined by `mnet` and `mmask`. If the source address of the host or hosts are within this range, they are allowed to attempt to log in. Any packet addressed to a switch IP interface with a source IP address outside this range is discarded silently.

Example: Assume that the `mnet` is set to 192.192.192.0 and the `mmask` is set to 255.255.255.128. This defines the following range of allowed IP addresses: 192.192.192.1 to 192.192.192.127.

- A host with a source IP address of 192.192.192.21 falls within the defined range and would be allowed to access the switch Management Processor.
- A host with a source IP address of 192.192.192.192 falls outside the defined range and is not granted access. To make this source IP address valid, you would need to shift the host to an IP address within the valid range specified by the `mnet` and `mmask` or modify the `mnet` to be 192.192.192.128 and the `mmask` to be 255.255.255.128. This would put the 192.192.192.192 host within the valid range allowed by the `mnet` and `mmask` (192.192.192.128-255).

NOTE – When the `mnet` and `mmask` Management Processor filter is applied, Routing Interface Protocol (RIP) updates received by the switch will be discarded if the source IP address of the RIP packet(s) falls outside the specified range. You can correct this by configuring static routes.

Secure Switch Management

Secure switch management is needed for environments that perform significant management functions across the Internet. The following are some of the functions for secure management:

- Authentication of remote administrators

  Authentication is the action of determining and verifying who the administrator is; it usually involves a name and a password. The password can be either a fixed password or a challenge-response query.

- Authorization of remote administrators

  Once an administrator has been authenticated, authorization is the action of determining what that user is allowed to do. Authorization does not merely provide yes or no answers but may also customize the service for a particular administrator.

- Encryption of management information exchanged between the remote administrator and the switch

  Examples of protocols to encrypt management information are SSH and SSL. Web OS supports the encryption of management information, using SSH, on Alteon 708 Web switches.

Authentication and Authorization Protocols

NOTE – While authentication and authorization (AA) protocols and servers are designed to authenticate remote dial-up users (in addition to authorizing remote access capabilities to users), this overview is focused on using the AA model to authenticate and authorize remote administrators for managing a switch.

The AA model is based on a client/server model. The remote access server (RAS), the switch, is a client to the back-end database server. A remote user (the remote administrator) interacts only with the RAS, not the back-end server and database.

Two prominent AA protocols used to control dial-up access into networks are Cisco's TACACS+ (Terminal Access Controller Access Control System) and Livingston Enterprise's RADIUS (Remote Authentication Dial-In User Service). Web OS supports only the RADIUS authentication method.

Requirements

The required components for authorization and authentication are listed below:

- A remote administrator
- The Web switch with authentication and authorization protocol support, acting as a client in the AA model
- A back-end authentication and authorization server that performs the following functions:
  - Authenticates remote administrators
  - Checks the remote administrator's authorization to access the switch
  - Optionally, tracks and logs the administrator's activity while logging on
- An AA database that contains information about authorized administrators and their specific capabilities and privileges

RADIUS Authentication and Authorization

RADIUS is an access server authentication, authorization, and accounting protocol used to secure remote access to networks and network services against unauthorized access.

RADIUS consists of three components:

- A protocol with a frame format that utilizes UDP over IP (based on RFC 2138)
- A centralized server that stores all the user authorization information
- A client, in this case, the switch

The operation of RADIUS authentication and authorization protocol is similar to the AA model described in the previous section. The switch, acting as the RADIUS client, will communicate to the RADIUS server to authenticate and authorize a remote administrator using the protocol definitions specified in RFC 2138. Transactions between the client and RADIUS

server are authenticated through the use of a shared secret, which is never sent over the network. In addition, the remote administrator passwords are sent encrypted between the RADIUS client (the switch) and the back-end RADIUS server.

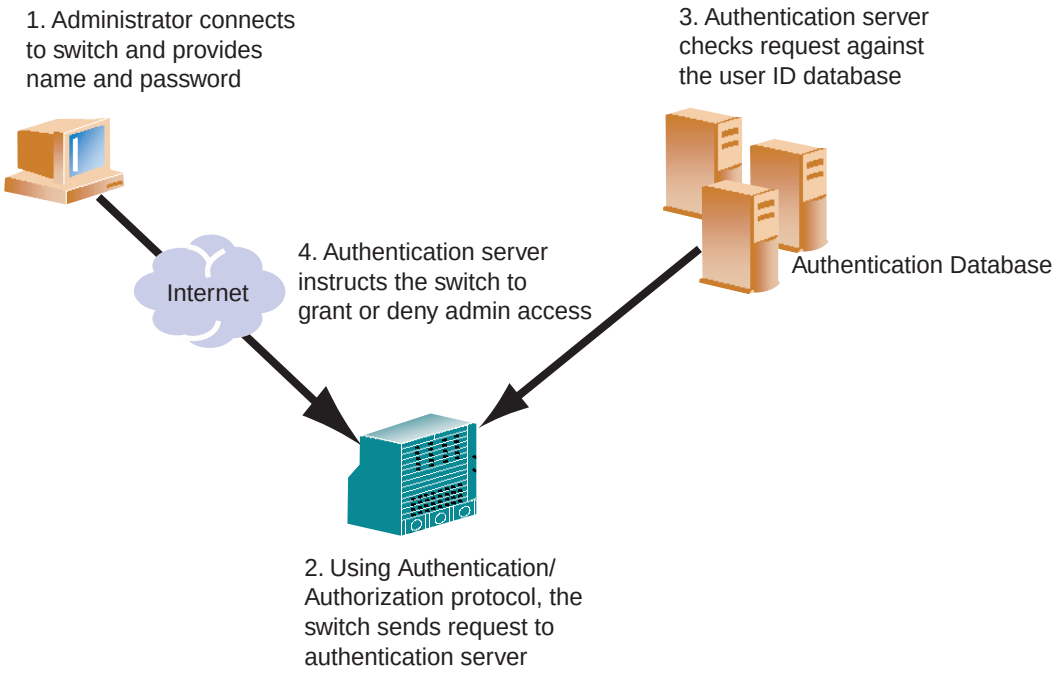


Figure 6-1 Authentication and Authorization: How It Works

RADIUS Authentication Features in Web OS

The following Radius Authentication features are supported in Web OS:

- Supports RADIUS client on the switch, based on the protocol definition in RFC 2138
- Supports enable/disable of RADIUS authentication and authorization
- The default disables the use of RADIUS for authentication and authorization.
- Allow RADIUS secret password up to 32 bytes and allows less than 16 octets
- Supports a *secondary authentication server* so that when the primary authentication server is unreachable, the switch can send client authentication requests to that secondary authentication server

  Use the **`/cfg/sys/radius/cur`** command to show the currently active RADIUS authentication server.

- Supports user-configurable RADIUS server retry and timeout values

  The parameters are:

  - Timeout value = 1-10 seconds
  - Retries = 1-3

    The switch will time out if it does not see response from the RADIUS server in 1-3 retries. The switch will also automatically retry to the RADIUS server before it declares the server is down.

- Supports user-configurable RADIUS application port

  The default is 1645/UDP as described in RFC 2138.

- Defines privileges for one or more specific users to access the switch at the RADIUS user database
- SecurID support, provided RADIUS server can do ACE/Server client proxy. The password is the PIN number, plus the token code of SecurID card.

Web Switch User Accounts

The following user accounts listed in Table 6-1 can be defined in the RADIUS server dictionary file:

Table 6-1 User Access Levels

User Account	Description and Tasks Performed	Password
User	The User has no direct responsibility for switch management. He/she can view all switch status information and statistics but cannot make any configuration changes to the switch.	`user`
SLB Operator	The SLB Operator manages Web servers and other Internet services and their loads. In addition to being able to view all switch information and statistics, the SLB Operator can enable/disable servers using the SLB operation menu.	`slboper`
Layer 4 Operator	The Layer 4 Operator manages traffic on the lines leading to the shared Internet services. This user currently has the same access level as the SLB operator. This level is reserved for future use, to provide access to operational commands for operators managing traffic on the line leading to the shared Internet services.	`l4oper`
Operator	The Operator manages all functions of the switch. In addition to SLB Operator functions, the Operator can reset ports or the entire switch.	`oper`
SLB Administrator	The SLB Administrator configures and manages Web servers and other Internet services and their loads. In addition to SLB Operator functions, the SLB Administrator can configure parameters on the SLB menus, with the exception of not being able to configure filters or bandwidth management.	`slbadmin`
Layer 4 Administrator	The Layer 4 Administrator configures and manages traffic on the lines leading to the shared Internet services. In addition to SLB Administrator functions, the Layer 4 Administrator can configure all parameters on the SLB menus, including filters and bandwidth management.	`l4admin`
Administrator	The super-user Administrator has complete access to all menus, information, and configuration commands on the switch, including the ability to change both the user and administrator passwords.	`admin`

When the user logs in, the switch authenticates his/her level of access by sending the RADIUS access request, that is the *client authentication request* to the RADIUS authentication server.

If the remote user is successfully authenticated by the authentication server, the switch will verify the *privileges* of the remote user and authorize the appropriate access. When both the primary and secondary authentication servers are not reachable, the administrator has an option to allow *backdoor* access via the console only or console and telnet access. The default is *disable* for telnet access and *enable* for console access.

All user privileges, other than those assigned to the Administrator, must be defined in the RADIUS dictionary. The file name of the dictionary is RADIUS vendor-dependent. The following user privileges are Alteon WebSystems proprietary definitions.

Table 6-2 Alteon WebSystems User Access Levels

User Name/Access	User-Service-Type	Value
User	*Vendor-supplied*	255
SLB Operator	*Vendor-supplied*	254
Layer 4 Operator	*Vendor-supplied*	253
Operator	*Vendor-supplied*	252
SLB Administrator	*Vendor-supplied*	251
Layer 4 Administrator	*Vendor-supplied*	250

Secure Shell and Secure Copy

Although a remote network administrator can manage the configuration of an Alteon Web switch via Telnet, this method does not provide a secure connection. Using Secure Shell (SSH) and Secure Copy (SCP), messages between a remote administrator and the switch are transmitted. Figure 6-1 on page 95 illustrates secure switch management.

NOTE – Secure Shell (SSH)/Secure Copy (SCP) features are supported on the Alteon 708 Web switches and can only be configured via the console port, using the command line interface. However, SCP `putcfg` and TFTP `getcfg` can also change the SSH/SCP configuration.When SSH is enabled, SCP is also enabled.

SSH is a protocol that enables a remote administrator to securely log into another computer over a network to execute management commands. All the data sent over the network using SSH is encrypted and secured. Using SSH gives administrators an alternate way to manage the switch, one that provides strong security.

SCP is typically used to securely copy files from one machine to another. SCP uses SSH for encryption of data on the network. On an Alteon 708, SCP is used to download and upload the switch configuration via secure channels.

The benefits of using SSH and SCP are listed below:

- Authentication of remote administrators

  Identifying Administrator using NAME/PASSWORD
- Authorization of remote administrators
  - Determining the permitted actions
  - Customizing management messages between the remote administrator and the switch
- Encryption of management messages between the remote admininistrator and the switch
- Secure copy support

The Web OS implementation of SSH is based on SSH version 1.5 and supports SSH-1.5-1.x.xx. SSH clients of other versions (especially version 2) will not be supported. The following SSH clients have been tested:

- SSH 1.2.23 and SSH 1.2.27 for Linux (freeware)
- SecureCRT 3.0.2 and SecureCRT 3.0.3 for Windows NT (Van Dyke Technologies, Inc.)
- F-Secure SSH 1.1 for Windows (Data Fellows)

NOTE – The maximum number of simultaneous Telnet/SSH/SCP connections is four. The `/cfg/sys/radius/telnet` command also applies to SSH/SCP connections.

Encryption of Management Messages

The supported encryption and authentication methods for both SSH and SCP are listed below:

Server Host Authentication:	Client RSA authenticates the switch at the beginning of every connection
Key Exchange:	RSA
Encryption:	3DES-CBC, DES
User Authentication:	Local password authentication, RADIUS, `SecurID` (via RADIUS, for SSH only—does not apply to SCP)

SCP Services

NOTE – Administrator privileges are required to perform SCP commands. Set SCP admin password (this password must be different from the admin password).

The following four SCP commands are supported in this service. These commands are entered using the command line interface on the client that is running the SCP application:

- `getcfg` is used to upload the switch's configuration to the remote host via SCP.
- `putcfg` is used to download the switch's configuration from a remote host to the switch; the `diff` command will be automatically executed at the end of `putcfg` to notify the remote client of the difference between the new and the current configurations.
- `putcfg_apply_save` saves the new configuration to the FLASH memory after `putcfg_apply` is done.

The `putcfg_apply` and `putcfg_apply_save` commands are provided because extra `apply` and `save` commands are usually required after a `putcfg`; however, an SCP session is not in an interactive mode at all.

RSA Host and Server Keys

To support the SSH server feature, two sets of RSA keys (host and server keys) are required. The host key is 1024 bits and is used to identify the switch. The server key is 768 bits and is used to make it impossible to decipher a captured session by breaking into the switch at a later time.

When the SSH server is first enabled and applied, the switch will automatically generate the host and server keys and then stores them into the FLASH memory.

NOTE – The Web switch will perform only one session of key/cipher generation at a time. Thus, an SSH/SCP client will not be able to log in if the switch is performing key generation at that time, or, if another client has logged in immediately prior. Also, key generation will fail if an SSH/SCP client is logging in at that time.

- To generate a host key:

```
>> # /cfg/sys/sshd/hkeygen
```

- To generate a server key:

```
>> # /cfg/sys/sshd/skeygen
```

Again, the host and server key are automatically stored into FLASH memory after they are generated.

NOTE – For security reasons, the SSHD menu options are available via the console port only and not via Telnet, SNMP, or WebUI.

When the switch reboots, it will retrieve the host and server keys from the FLASH. If these two keys are not available in the flash, and if the SSH server feature is enabled, the switch will automatically generate them during the system reboot.

The switch can also automatically regenerate the RSA server key. To set the interval of RSA server key autogeneration, use this command:

```
>> # /cfg/sys/sshd/intrval <number of hours (0-24)>
```

where *number of hours* must be between 0–24, and a value of 0 denotes that the RSA server key autogeneration is disabled. When the value is greater than zero, the switch will autogenerate the RSA server key every *number of hours*; however, RSA server key generation will be skipped if the switch is busy doing other key or cipher generation when the timer expires.

Radius Authentication

SSH/SCP is integrated with RADIUS authentication. After the RADIUS server is enabled in the switch, all subsequent SSH authentication requests will be redirected to the specified RADIUS servers for authentication. The redirection is transparent to the SSH clients.

Secure ID Support

SSH/SCP can also work with SecurID, a token card-based authentication method. The use of SecurID requires the interactive mode during login, which is not provided by the SSH connection.

NOTE – There is no SNMP or Browser-Based Interface (BBI) support for SecurID because the SecurID server, ACE, is a one-time password authentication and requires an interactive session.

To log in using SSH without difficulties, you need to use a special username, “ace,” to log in and bypass the SSH authentication. After an SSH connection is established, you will then be prompted to enter the username and password (the SecurID authentication is being performed now). You will need to provide your actual username and the token in your SecurID card as a regular Telnet user would do in order to log in.

To use SCP, you need to use SCP-only administrator’s password (that is, the `scpadm` option under the `/cfg/sys/sshd/` menu) to bypass the checking of SecurID. Alternately, you can configure a regular administrator with a fixed password in the RADIUS server if it can be supported. A regular administrator with a fixed password in the RADIUS server can perform both SSH and SCP without additional authentication required.

An SCP-only administrator’s password is typically used when SecurID is used. For example, it can be used in an automation program (in which the tokens of SecurID are not available) to back up (download) the switch configurations each day.

NOTE – The SCP-only administrator’s password must be different from the regular administrator’s password. If the two passwords are the same, the administrator using that password will not be allowed to log in as an SSH user because the switch will recognize him as the SCP-only administrator and only allow the administrator access to SCP commands.

Configuring SSH/SCP

NOTE – SSH/SCP parameters can be configured only via the console port, using the Command Line Interface (CLI). The switch SSH daemon uses TCP port 22 only and is not configurable.

To enable or disable the SSH/SCP feature, use one of these commands:

```
>> # /cfg/sys/sshd/on                    (Enable SSH/SCP)
>> # /cfg/sys/sshd/off                   (Disable SSH/SCP)
```

To set the interval of RSA server key autogeneration, use this command:

```
>> # /cfg/sys/sshd/intrval <number of hours>
```

where number of hours must be between 0–24, and a value of 0 denotes that RSA server key autogeneration is disabled. When *number of hours* is greater than 0, the switch will autogenerate the RSA server key every *interval specified*; however, RSA server key generation will be skipped if the switch is busy doing other key or cipher generation when the timer expires.

To enable or disable the SCP apply and save (i.e., SCP `putcfg_apply` and `putcfg_apply_save` commands), use these commands:

```
>> # /cfg/sys/sshd/ena                   (Enable SSH/SCP apply and save)
>> # /cfg/sys/sshd/dis                   (Disable SSH/SCP apply and save)
```

To view the current SSH/SCP-related configuration, use this command:

```
>> # /cfg/sys/sshd/cur
```

To view the difference between the new and current configurations, use this command:

```
>> # diff
```

NOTE – If SSH/SCP is enabled and an `apply` command is issued, the switch will automatically generate the RSA host and server keys if they are not available. It will take several minutes to complete this process.

To save the current configuration to flash, use this command:

```
>> # save
```

Usually, there will be no need to generate manually the RSA host and server keys. However, you may still do so by using the following commands:

```
>> # /cfg/sys/sshd/hkeygen          (Generates the host key)
>> # /cfg/sys/sshd/skeygen          (Generates the server key)
```

NOTE – These two commands will take effect immediately without the need of an `apply` command being issued.

Some Supported Client Commands

NOTE – Up to four simultaneous Telnet/SSH/SCP connections are supported on a switch.

- To log in to the switch:

  ssh <*switch IP*>or **ssh -l** <*user name*> <*switch IP*>

- To download the switch configuration using SCP:

  scp <*switch IP*>**:getcfg** <*local filename*>

- To upload the configuration to the switch:

  scp <*local filename*> <*switch IP*>**:putcfg**

Some examples are listed below:

```
>> # ssh 205.178.15.157
>> # ssh -l dleu 205.178.15.157
>> # scp 205.178.15.157:getcfg ad4.cfg
>> # scp ad4.cfg 205.178.15.157:putcfg
```

where *205.178.15.157* is the IP address of the switch.

Please also note that `apply` and `save` commands are still needed after the last command (`scp ad4.cfg 205.178.15.157:putcfg`) is issued. Or, instead, you can use the following commands.

```
>> # scp ad4.cfg 205.178.15.157:putcfg_apply
>> # scp ad4.cfg 205.178.15.157:putcfg_apply_save
```

Part 2: Web Switching Fundamentals

Web traffic is different from classic IP-based LAN data traffic. This section discusses Web switching basics, which consists primarily of optimizing Web-based IP traffic or HTTP traffic.

- Server Load Balancing
- Filtering
- Application Redirection
- Health Checking
- High Availability

050156A, June 2001

Alteon WebSystems
Web Speed for e-Business

CHAPTER 7

Server Load Balancing

This chapter describes how to configure an Alteon 708 for server load balancing (SLB). The following sections in this chapter describe how to configure and use the optional Layer 4 software for Server Load Balancing (SLB):

- "Understanding Server Load Balancing" on page 108. This section discusses the benefits of server load balancing and how it works.
- "Network Topology Requirements" on page 110. This section provides key requirements to consider before deploying server load balancing.
- "Configuring Server Load Balancing" on page 112. This section provides a step-by-step procedure on how to configure your switch for server load balancing.
- "Additional SLB Options" on page 117. This section discusses other configuration options that can be used to customize your Web switch.
- "Direct Client Access to Real Servers" on page 126. This section describes how to configure the Web switch to allow some clients to have direct access to the real servers.

For information on activating the SLB feature, see your *Web OS 7.2 Command Reference*.

Understanding Server Load Balancing

Benefits

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

- Increased efficiency for server utilization and network bandwidth

  With SLB, your Alteon 708 Web switch is aware of the shared services provided by your server pool and can then balance user session traffic among the available servers. Important session traffic gets through more easily, reducing user competition for connections on overutilized servers. For even greater control, traffic is distributed according to a variety of user-selectable rules.

- 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 Web site 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 Server Load Balancing 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 overutilized. 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, overutilization of key servers often happens in networks where other servers are actually underutilized.

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

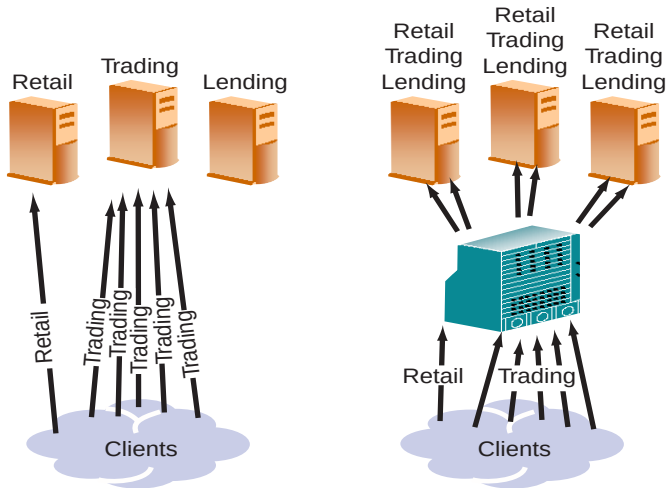


Figure 7-1 Traditional Versus SLB Network Configurations

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 acts 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 up to 1024 virtual servers on the switch, each distributing up to eight different services (up to a total of 4000).

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 Requirements

When deploying SLB, consider these key requirements:

- In standard SLB, all client requests to a virtual IP address and all responses from the real servers must pass through the switch as shown in Figure 7-2. If there is a path between the client and the real servers that does not pass through the switch, the Web switch can be configured to proxy requests in order to guarantee that responses use the correct path (see "Proxy IP Addresses for Complex SLB Networks" on page 123).

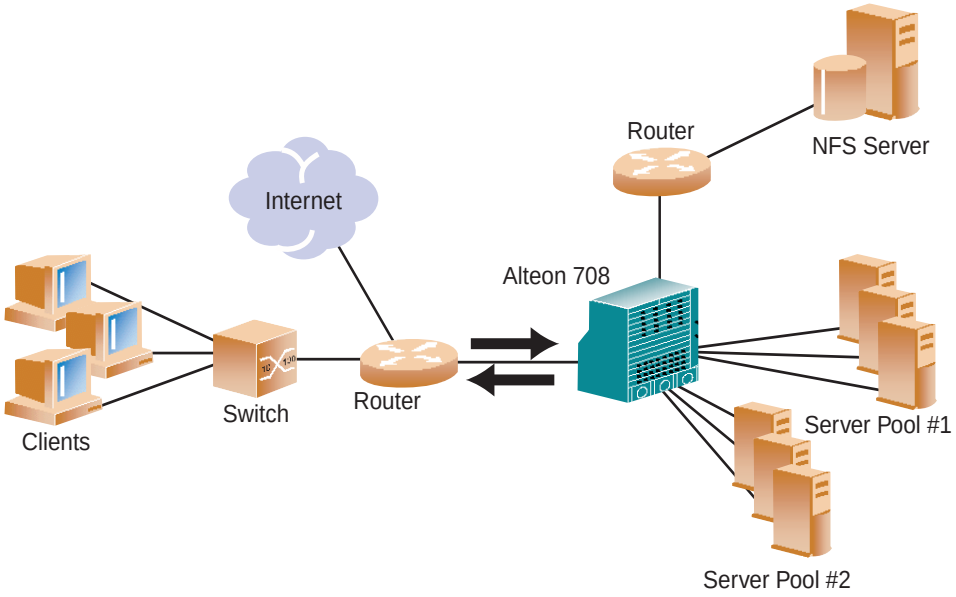


Figure 7-2 SLB Client/Server Traffic Routing

- 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 go 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 use the `minmisses` or `hash` metrics (see "Metrics for Real Server Groups" on page 118).

- 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 that are configured to process client request traffic provide address translation from the virtual IP to the real server IP address.
  - Layer 4 server processing: Ports that are 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:

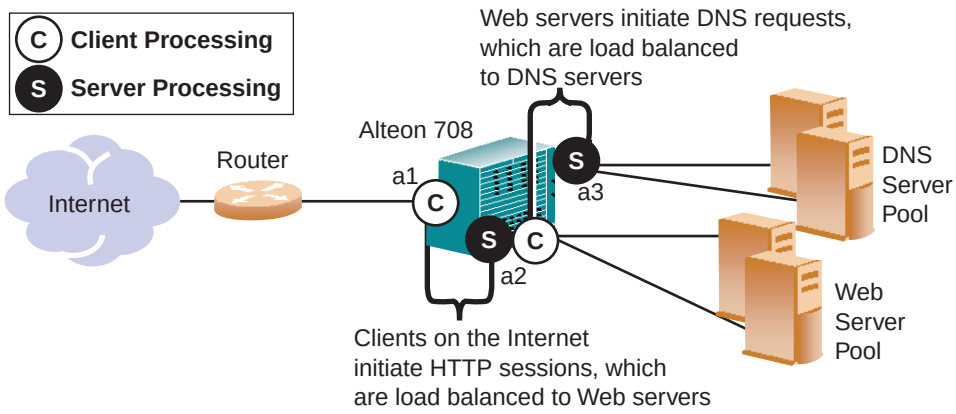


Figure 7-3 Example Network for Client/Server Port Configuration

In Figure 7-3, 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.

Configuring Server Load Balancing

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.

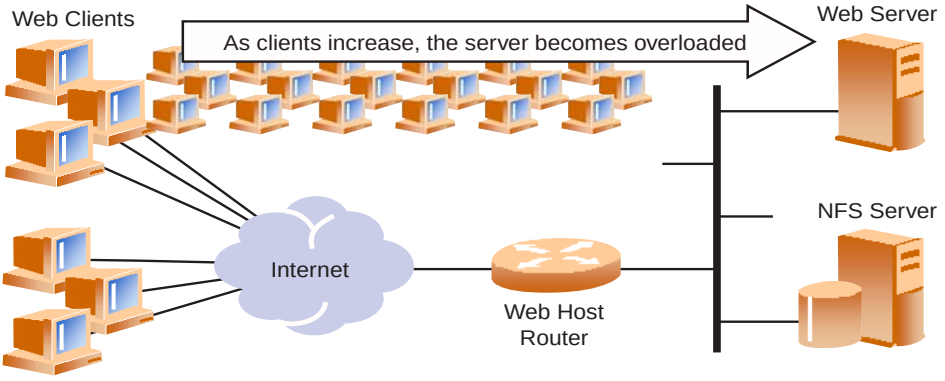


Figure 7-4 Web Hosting Configuration Without SLB

Such a company has three primary needs:

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

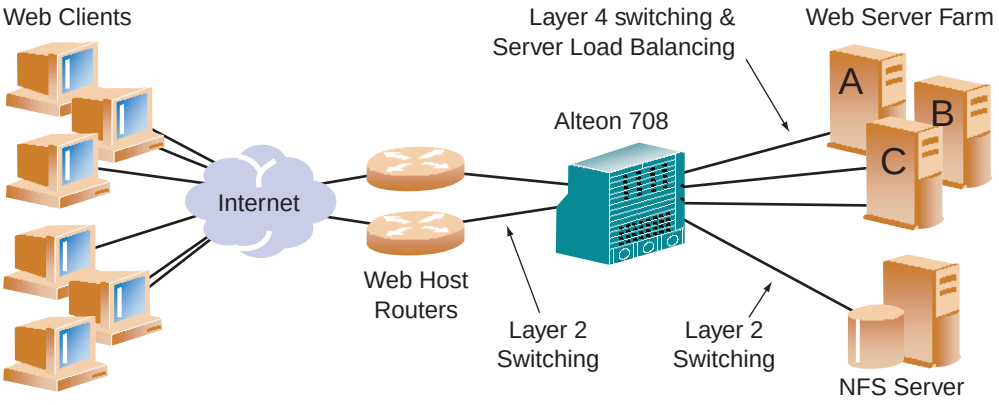


Figure 7-5 Web Hosting with SLB Solutions

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

- Reliability is increased by providing multiple paths from the clients to the Web switch and by accessing a pool of servers with 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.

In the following procedure, many of the SLB options are left to their default values. See "Additional SLB Options" on page 117 for more options.

The following is required prior to configuration:

- Connect to the switch command line interface as the administrator.
- Enable the SLB software key.

NOTE – For details about any of the menu commands described in this example, see your *Web OS 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 IP routing 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 Requirements" on page 110.

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

Table 7-1 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 117).

---

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

   The switch must have an IP route to all of the real servers that 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. **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. **Define a real server group and add three real services to the 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. 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 the 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 Command Reference*.

6. Define the port settings.

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

Table 7-2 Web Host Example: Port Usage

Port	Host	L4 Processing
a1	Server A serves SLB requests.	Server
a2	Server B serves SLB requests.	Server
a3	Server C serves SLB requests.	Server
a4	Back-end NFS server provides centralized Web content for all three real servers from the NFS server. This port does not require Web switching features.	None
a5	Client router A connects the switch to the Internet where client requests originate.	Client
a6	Client router B 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. **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 the appropriate changes.

8. **Save your new configuration changes.**

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

9. **Verify the SLB information.**

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

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

Additional SLB Options

In the previous section ("Configuring Server Load Balancing" on page 112), many of the SLB options are left to their default values. The following configuration options can be used to customize SLB on your Web switch:

- IP Address Ranges Using imask
- Health Checks for Real Servers
- Metrics for Real Server Groups
- Weights for Real Servers
- Connection Time-outs for Real Servers
- Maximum Connections for Real Servers
- Backup/Overflow Servers
- Mapping Virtual Server Ports to Real Server Ports
- Direct Server Return
- Proxy IP Addresses for Complex SLB Networks
- Disabling Real Servers
- IP Server Load Balancing

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 `minmisses` (minimum misses), `hash`, `leastconns` (least connections), and `roundrobin` are 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)
```

Minimum Misses

The `minmisses` 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 is 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 is sent to the same server. This metric is useful for applications where client information must be retained on the server between sessions. With this metric, server load becomes most evenly balanced as the number of active clients with different source or destination addresses increases.

When selecting a server, the switch calculates 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) because 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 option is useful for applications where client information must be retained between sessions.
- For FWLB, both the source and destination IP addresses are used to ensure that the two unidirectional flows of a given session are 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.

NOTE – 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.

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; fastest servers typically get the most connections over time because of their ability to accept, process, and shut down connections faster than slower servers.

Round Robin

With the `roundrobin` metric, new connections are issued to each server in turn; that is, 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 larger share of connections. 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 number of minutes)
```

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

Maximum Connections for Real Servers

You can set the number of open connections each real server is allowed to handle for SLB. 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 from approximately 500 HTTP connections for slower servers to 1500 for quicker, multiprocessor servers. The appropriate value also depends on the duration of each session and 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 benchmark 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 back up 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 back up. 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 Server Ports to Real Server Ports

In addition to providing direct real server access in some situations (see "Port Mapping" on page 127), 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 (DAM) 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 126.

Direct Server Return

The Direct Server Return (DSR) feature allows the server to respond directly to the client. This capability is useful for sites where large amounts of data are flowing from servers to clients, such as with content providers or portal sites that typically have asymmetric traffic patterns.

NOTE – DSR requires that the server must be set up to receive frames that have an IP destination address that is equal to the virtual server IP address(es).

The following sequence of steps is illustrated in Figure 7-6:

1. **A client request is forwarded to the Web switch.**
2. **Because only MAC addresses are substituted, the switch forwards the request to the best server, based on the configured load-balancing policy.**

3. **The server responds directly to the client, bypassing the switch, using the virtual IP address as the IP source address.**

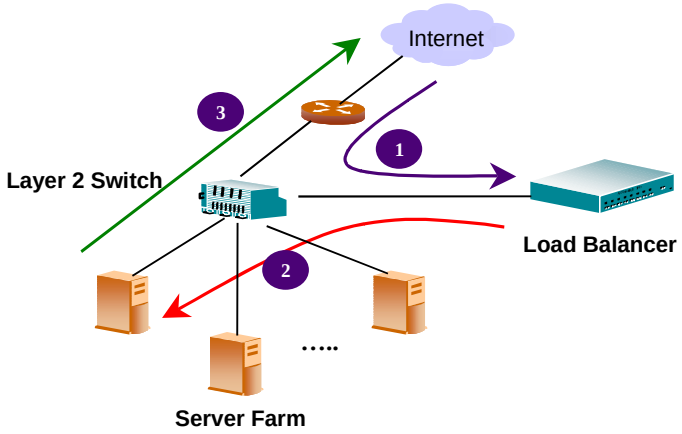


Figure 7-6 Direct Server Return

To set up direct server return, use the following commands:

```
>> # /cfg/slb/port <port number>/submac ena
>> # /cfg/slb/virt <virtual server number>/service <service number>/nonat ena
```

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. 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 126).

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 reconfigured to disable server processing. Refer to the Table 7-2 on page 115, with the following revised port configuration:

Table 7-3 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 NFS server. This port does not require Web switching.	None
a5	Client router A: Connects the switch to the Internet, where all client requests originate.	Client
a6	Client router B: Connects the switch to the Internet, where all client requests originate.	Client

The following commands are used to disable server processing on ports a1-a3:

```
>> # /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* port requires a proxy IP address. Each proxy IP address must be unique on your network. The following commands are used to configure client proxies.

```
>> # /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. You can configure up to 7 proxy IP addresses on an Alteon 708.

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

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

Disabling Real Servers

If you need to reboot a server, you must make sure that new sessions are not sent to the real server and that old sessions are not discarded. When the session count gets to zero, you may shut down the server. Use the following command, to disable real servers:

```
>> # /oper/slb/dis <real server number>
```

IP Server Load Balancing

IP load balancing allows you to configure your switch for server load balancing based on client's IP address only. Typically, the client IP address is used with the client port number to produce a session identifier. When the layer 3 option is enabled, the switch uses only the client IP address as the session identifier.

To configure the switch for IP load balancing,

```
>> # /cfg/slb/virt <virtual server number>
>> Virtual Server 1# layr3 ena
>> Virtual Server 1# service ip
>> Virtual Server 1 ip Service# group <group # >
```

Configure virtual IP address on loopback for IP load balancing of DNS.

Direct Client Access to Real Servers

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

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

Direct Access Mode

When Direct Access Mode (DAM) (`/cfg/slb/adv/direct`) is enabled on a switch, any client can communicate with any real server's load-balanced service. Also, with DAM enabled, 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.

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 "Proxy IP Addresses for Complex SLB Networks" on page 123). 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, the client can access each real server directly using the real server's IP address. The switch port must be connected to the real server and client processing must be disabled (see the `server` and `client` options under the command, `/cfg/slb/port` in your *Web OS Command Reference*).

SLB is still accessed using the virtual server IP address.

Port Mapping

When SLB is used without proxy IP addresses and without 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 remapped 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 TCP port 8000 on the real server port.

In the Figure 7-7, 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 as the real server's IP address.

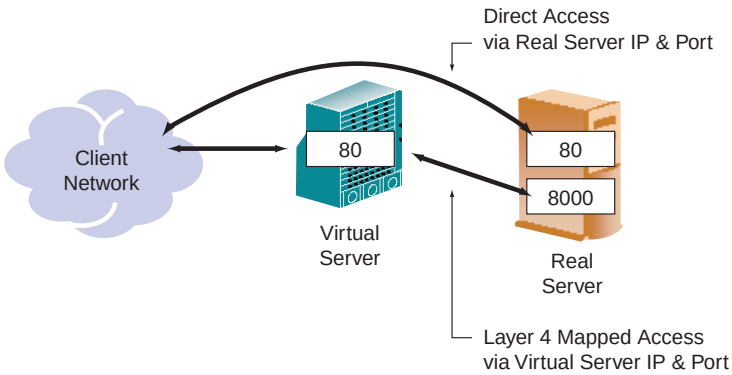


Figure 7-7 Mapped and Nonmapped Server Access

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/adv`), 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 is 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 that is allowed direct real server access only.

CHAPTER 8
Filtering

This chapter provides a conceptual overview of filters and configuration examples showing how filters can be used to ensure network security and redirect traffic. The following topics are discussed in this chapter:

- "Filtering Overview" on page 130. This section describes the benefits and filtering criteria to allow for extensive filtering at the IP and TCP/UDP levels.
  - "Filtering Benefits" on page 130
  - "Filtering Criteria" on page 130
  - "Defining IP Address Ranges for Filters" on page 132
  - "Stacking Filters" on page 132
  - "Overlapping Filters" on page 133
  - "The Default Filter" on page 133
  - "Optimizing Filter Performance" on page 134
  - "Filter Logs" on page 134
- "Security Example" on page 135. This section provides an example of configuring filters to provide the best security. It is generally recommended that you configure filters to deny all traffic except for those services that you specifically wish to allow.
- "Network Address Translation Examples" on page 141. This section provides two examples: Internal client access to the internet and external client access to the server.

Filtering Overview

Alteon WebSystems switches are used to deliver content efficiently and secure your servers from unauthorized intrusion, probing, and Denial-Of-Service (DoS) attacks. The capabilities built into Web OS allow for extensive filtering at the IP and TCP/UDP levels.

Filtering Benefits

Layer 3 (IP) and Layer 4 (application) filtering give the network administrator a powerful tool with the following benefits:

- Increased security for server networks

  Filters can be configured to allow or deny traffic according to various IP address, protocol, and Layer 4 port criteria. This gives the administrator control over the types of traffic permitted through the switch. Any filter can be optionally configured to generate `syslog` messages for increased security visibility.

- Used to map the source or destination IP addresses and the port

  Generic Network Address Translation (NAT) can be used to map the source or destination IP address and the port of private network traffic, to and from an advertised network IP address and port.

Filtering Criteria

Up to 1024 filters can be configured on the switch. Each filter can be set to allow, deny, redirect, or translate traffic based on any combination of the following criteria:

- `sip`: source IP address or range (see page 132)
- `dip`: destination IP address or range (`dip` and `dmask`)
- `proto`: protocol number or name as shown in Table 8-1

Table 8-1 Well-Known Protocol Types

Number	Protocol Name
6	tcp
17	udp
89	ospf
112	vrrp

- `sport`: TCP/UDP application or source port as shown in Table 8-2, or source port range (such as 31000-33000)

Table 8-2 Well-Known Application Ports

Number	TCP/UDP Application	Number	TCP/UDP Application	Number	TCP/UDP Application
20	ftp-data	70	gopher	161	snmp
21	ftp	79	finger	162	snmptrap
22	ssh	80	http	179	bgp
23	telnet	109	pop2	194	irc
25	smtp	110	pop3	220	imap3
37	time	111	sunrpc	389	ldap
42	name	119	nntp	443	https
43	whois	123	ntp	520	rip
53	domain	143	imap	554	rtsp
69	tftp	144	news	1985	hsrp

- `dport`: TCP/UDP application or destination port as shown in Table 8-2, or destination port range (such as 31000-33000)
- `invert`: reverse the filter logic in order to activate the filter whenever the specified conditions are *not* met.

Using these filter criteria you can create a single filter that blocks external Telnet traffic to your main server except from a trusted IP address. Another filter could warn you if FTP access is attempted from a specific IP address. Another filter could redirect all incoming e-mail traffic to a server where it can be analyzed for spam. The options are nearly endless.

Defining IP Address Ranges for Filters

You can specify a range of IP addresses for filtering both the source and/or destination IP address for traffic. When a range of IP addresses is needed, the source IP (SIP) address or destination IP (DIP) address defines the base IP address in the desired range, and the `smask` (source) or `dmask` (destination) is the mask that 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, the switch could be configured 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 8-3 Filtering IP Address Ranges

Filter	Internet Address Range	dip	dmask
#1	0.0.0.0 - 127.255.255.255	0.0.0.0	128.0.0.0
#2	128.0.0.0 - 255.255.255.255	128.0.0.0	128.0.0.0

Stacking Filters

Stacking filters are assigned and enabled on a per-port basis. Each filter can be used by itself or in combination with any other filter on any given switch port. The filters are numbered 1 through 1024. When multiple filters are stacked together on a port, the filter's number determines its order of precedence: the filter with the lowest number is checked first. When traffic is encountered at the switch port, if the filter matches, its configured action takes place and the rest of the filters are ignored. If the filter criteria doesn't match, the next filter is tried.

As long as the filters do not overlap, you can improve filter performance by making sure that the most heavily utilized filters are applied first. For example, consider a filter system where the Internet is divided according to destination IP address:

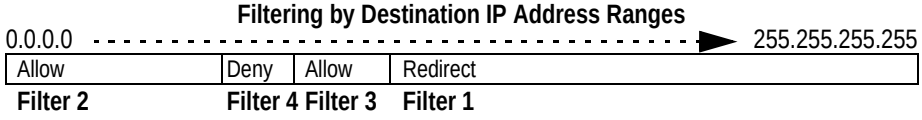


Figure 8-1 Assigning Filters According to Range of Coverage

Assuming that traffic is distributed evenly across the Internet, the largest area would be the most utilized and is assigned to Filter 1. The smallest area is assigned to Filter 4.

Overlapping Filters

Filters are permitted to overlap, although special care should be taken to ensure the proper order of precedence. When overlapping filters are present, the more specific filters (those that target fewer addresses or ports) should be applied before the generalized filters.

Example:

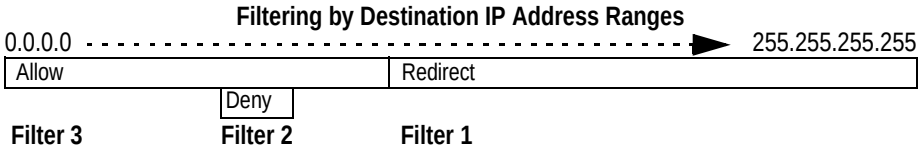


Figure 8-2 Assigning Filters to Overlapping Ranges

In this example, the `deny` filter must be processed prior to the `allow` filter. If the `allow` filter was permitted to take precedence, the `deny` filter could never be triggered.

The Default Filter

Before filtering can be enabled on any given port, a default filter should be configured. This filter handles any traffic not covered by any other filter. All the criteria in the default filter must be set to the full range possible (`any`). For example:

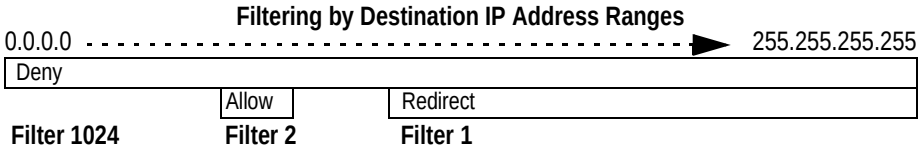


Figure 8-3 Assigning a Default Filter

In this example, the default filter is defined as filter 1024 in order to give it the lowest order of precedence. All matching criteria in filter 1024 is set to the `any` state. If no other filter acts on the traffic, filter 1024 handles it, denying and logging unwanted traffic.

```
>> # /cfg/slb/filt 1024                    (Select the default filter)
>> Filter 1024# sip any                     (From any source IP addresses)
>> Filter 1024# dip any                     (To any destination IP addresses)
>> Filter 1024# proto any                   (For any protocols)
>> Filter 1024# action deny                 (Deny matching traffic)
>> Filter 1024# enable                      (Enable the default filter)
>> Filter 1024# adv                         (Select the advanced menu)
>> Filter 1024 Advanced# log enable         (Log matching traffic to syslog)
```

Default filters are recommended (but not required) when configuring filters for IP traffic control and redirection. Using default filters can increase session performance but takes some of the session binding resources. If you experience an unacceptable number of binding failures as shown in the Server Load Balancing Maintenance Statistics (`/stats/slb/maint`), you may wish to remove some of the default filters.

Optimizing Filter Performance

Filter efficiency can be increased by placing filters that are used most often near the beginning of the filtering list. It is recommended to number filters in small increments (for example: 5, 10, 15, 20, etc.) so that filters could be easily inserted between others in the list, if required.

Filter Logs

To provide enhanced troubleshooting and session inspection capability, packet source and destination IP addresses are included in filter log messages. Filter log messages are generated when a Layer 3/Layer 4 filter is triggered and has logging enabled. The messages are output to the console port, system host log (`syslog`), and the Web-based interface message window.

Example: A network administrator has noticed a significant number of ICMP frames on one portion of the network and wants to determine the specific sources of the ICMP messages. The administrator uses the CLI to create and apply the following filter:

```
>> # /cfg/slb/filt 15                  (Select filter 15)
>> Filter 15# sip any                  (From any source IP address)
>> Filter 15# dip any                  (To any destination IP address)
>> Filter 15# action allow             (Allows matching traffic to pass)
>> Filter 15# proto icmp               (For the ICMP protocol)
>> Filter 15# ena                      (Enable the filter)
>> Filter 15# adv/log enable           (Create a log entry when matched)
>> Filter 15# /cfg/slb/port b7         (Select a switch port to filter)
>> SLB port B7# add 15                 (Add the filter to the switch port)
>> SLB port B7# filt ena               (Enable filtering on the switch port)
>> SLB port B7# apply                  (Apply the configuration changes)
>> SLB port B7# save                   (Save the configuration changes)
```

When applied to one or more switch ports, this simple filter rule will produce log messages that show when the filter is triggered, and what the IP source and destination addresses were for the ICMP frames traversing those ports.

Example: Filter log message output is shown below, displaying the filter number, port, source IP address, and destination IP address:

```
slb: filter 15 fired on port b7, 206.118.93.110 -> 20.10.1.10
```

Security Example

Consider the following sample network:

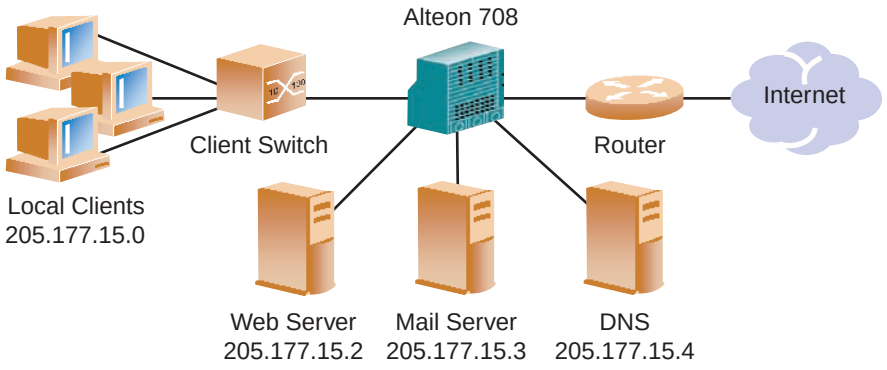


Figure 8-4 Security Topology Example

In this example, the network is made of local clients on a collector switch, a Web server, a mail server, a domain name server, and a connection to the Internet. All the local devices are on the same subnet.

For best security, it is generally recommended that you configure filters to deny all traffic except for those services that you specifically wish to allow. In this example, the administrator wishes to install basic security filters to allow only the following traffic:

- External HTTP access to the local Web server
- External SMTP (mail) access to the local mail server
- Local clients browsing the World Wide Web
- Local clients using Telnet to access sites outside the intranet
- DNS traffic

All other traffic will be denied and logged.

NOTE – Since IP address and port information can be manipulated by external sources, filtering does not replace the necessity for a well-constructed network firewall.

Example Configuration for the Security Solution

Prior to configuration, you must be connected to the switch command line interface as the administrator.

In this example, all filters are applied only to the switch port that connects to the Internet. If intranet restrictions are required, filters can be placed on switch ports connecting to local devices.

Also, filtering is not limited to the few protocols and TCP or UDP applications shown in this example. See the *Web OS 7.2 Command Reference* for a list of other well-known protocols and services.

1. **Assign an IP address to each of the network devices.**

   For this example, the network devices have the following IP addresses on the same IP subnet:

   Table 8-4 Web Cache Example: Real Server IP Addresses

Network Device	IP address
Local Subnet	205.177.15.0 - 205.177.15.255
Web Server	205.177.15.2
Mail Server	205.177.15.3
Domain Name Server	205.177.15.4

2. **On the switch, create a default filter that will deny and log unwanted traffic.**

   The default filter is defined as filter 1024 in order to give it the lowest order of precedence:

```
>> # /cfg/slb/filt 1024                    (Select the default filter)
>> Filter 1024# sip any                     (From any source IP addresses)
>> Filter 1024# dip any                     (To any destination IP addresses)
>> Filter 1024# proto any                   (For any protocols)
>> Filter 1024# action deny                 (Deny matching traffic)
>> Filter 1024# enable                      (Enable the default filter)
>> Filter 1024# adv                         (Select the advanced menu)
>> Filter 1024 Advanced# log enable         (Log matching traffic to syslog)
```

3. **Create a filter that will allow external HTTP requests to reach the Web server.**

   The filter must recognize and allow TCP traffic with the Web server's destination IP address and HTTP destination port:

```
>> Filter 1024# /cfg/slb/filt 1          (Select the menu for filter 1)
>> Filter 1# sip any                     (From any source IP address)
>> Filter 1# dip 205.177.15.2            (To Web server dest. IP address)
>> Filter 1# dmask 255.255.255.255       (Fill mask for exact dest. address)
>> Filter 1# proto tcp                   (For TCP protocol traffic)
>> Filter 1# sport any                   (From any source port)
>> Filter 1# dport http                  (To an HTTP destination port)
>> Filter 1# action allow                (Allow matching traffic to pass)
>> Filter 1# ena                         (Enable the filter)
```

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

4. **Create a pair of filters to allow incoming and outgoing mail to and from the mail server.**

   Filter 2 allows incoming mail to reach the mail server, and filter 3 allows outgoing mail to reach the Internet:

```
>> Filter 1# ../filt 2                   (Select the menu for filter 2)
>> Filter 2# sip any                     (From any source IP address)
>> Filter 2# dip 205.177.15.3            (To mail server dest. IP address)
>> Filter 2# dmask 255.255.255.255       (Fill mask for exact dest. address)
>> Filter 2# proto tcp                   (For TCP protocol traffic)
>> Filter 2# sport any                   (From any source port)
>> Filter 2# dport smtp                  (To a SMTP destination port)
>> Filter 2# action allow                (Allow matching traffic to pass)
>> Filter 2# ena                         (Enable the filter)
>> Filter 2# ../filt 3                   (Select the menu for filter 3)
>> Filter 3# sip 205.177.15.3            (From mail server source IP address)
>> Filter 3# smask 255.255.255.255       (Fill mask for exact source address)
>> Filter 3# dip any                     (To any destination IP address)
>> Filter 3# proto tcp                   (For TCP protocol traffic)
>> Filter 3# sport smtp                  (From a SMTP port)
>> Filter 3# dport any                   (To any destination port)
>> Filter 3# action allow                (Allow matching traffic to pass)
>> Filter 3# ena                         (Enable the filter)
```

5. **Create a filter that will allow local clients to browse the Web.**

   The filter must recognize and allow TCP traffic to reach the local client destination IP addresses if originating from any HTTP source port:

```
>> Filter 3# ../filt 4                  (Select the menu for Filter 4)
>> Filter 4# sip any                    (From any source IP address)
>> Filter 4# dip 205.177.15.0           (To base local network dest. address)
>> Filter 4# dmask 255.255.255.0        (For entire subnet range)
>> Filter 4# proto tcp                  (For TCP protocol traffic)
>> Filter 4# sport http                 (From any source HTTP port)
>> Filter 4# dport any                  (To any destination port)
>> Filter 4# action allow               (Allow matching traffic to pass)
>> Filter 4# ena                        (Enable the filter)
```

6. **Create a filter that will allow local clients to Telnet anywhere outside the local intranet.**

   The filter must recognize and allow TCP traffic to reach the local client destination IP addresses if originating from a Telnet source port:

```
>> Filter 4# ../filt 5                  (Select the menu for Filter 5)
>> Filter 5# sip any                    (From any source IP address)
>> Filter 5# dip 205.177.15.0           (To base local network dest. address)
>> Filter 5# dmask 255.255.255.0        (For entire subnet range)
>> Filter 5# proto tcp                  (For TCP protocol traffic)
>> Filter 5# sport telnet               (From a Telnet port)
>> Filter 5# dport any                  (To any destination port)
>> Filter 5# action allow               (Allow matching traffic to pass)
>> Filter 5# ena                        (Enable the filter)
```

7. **Create a series of filters to allow Domain Name System (DNS) traffic.**

   DNS traffic requires four filters; one pair is needed for UDP traffic (incoming and outgoing) and another pair for TCP traffic (incoming and outgoing).

For UDP:

```
>> Filter 5# ../filt 6                    (Select the menu for Filter 6)
>> Filter 6# sip any                      (From any source IP address)
>> Filter 6# dip 205.177.15.4             (To local DNS Server)
>> Filter 6# dmask 255.255.255.255        (Fill mask for exact dest. address)
>> Filter 6# proto udp                    (For UDP protocol traffic)
>> Filter 6# sport any                    (From any source port)
>> Filter 6# dport domain                 (To any DNS destination port)
>> Filter 6# action allow                 (Allow matching traffic to pass)
>> Filter 6# ena                          (Enable the filter)
>> Filter 6# ../filt 7                    (Select the menu for Filter 7)
>> Filter 7# sip 205.177.15.4             (From local DNS Server)
>> Filter 7# smask 255.255.255.255        (Fill mask for exact source address)
>> Filter 7# dip any                      (To any destination IP address)
>> Filter 7# proto udp                    (For UDP protocol traffic)
>> Filter 7# sport domain                 (From a DNS source port)
>> Filter 7# dport any                    (To any destination port)
>> Filter 7# action allow                 (Allow matching traffic to pass)
>> Filter 7# ena                          (Enable the filter)
```

Similarly, for TCP:

```
>> Filter 7# ../filt 8                    (Select the menu for Filter 8)
>> Filter 8# sip any                      (From any source IP address)
>> Filter 8# dip 205.177.15.4             (To local DNS Server)
>> Filter 8# dmask 255.255.255.255        (Fill mask for exact dest. address)
>> Filter 8# proto tcp                    (For TCP protocol traffic)
>> Filter 8# sport any                    (From any source port)
>> Filter 8# dport domain                 (To any DNS destination port)
>> Filter 8# action allow                 (Allow matching traffic to pass)
>> Filter 8# ena                          (Enable the filter)
>> Filter 8# ../filt 9                    (Select the menu for Filter 9)
>> Filter 9# sip 205.177.15.4             (From local DNS Server)
>> Filter 9# smask 255.255.255.255        (Fill mask for exact source address)
>> Filter 9# dip any                      (To any destination IP address)
>> Filter 9# proto tcp                    (For TCP protocol traffic)
>> Filter 9# sport domain                 (From a DNS source port)
>> Filter 9# dport any                    (To any destination port)
>> Filter 9# action allow                 (Allow matching traffic to pass)
>> Filter 9# ena                          (Enable the filter)
```

8. **Assign the filters to the switch port that connects to the Internet.**

```
>> Filter 9# /cfg/slb/port b5          (Select the SLB port b5 to the Internet)
>> SLB Port B5 # add 1                 (Add filter 1 to port b5)
>> SLB Port B5 # add 2                 (Add filter 2 to port b5)
>> SLB Port B5 # add 3                 (Add filter 3 to port b5)
>> SLB Port B5 # add 4                 (Add filter 4 to port b5)
>> SLB Port B5 # add 5                 (Add filter 5 to port b5)
>> SLB Port B5 # add 6                 (Add filter 6 to port b5)
>> SLB Port B5 # add 7                 (Add filter 7 to port b5)
>> SLB Port B5 # add 8                 (Add filter 8 to port b5)
>> SLB Port B5 # add 9                 (Add filter 9 to port b5)
>> SLB Port B5 # add 1024              (Add the default filter to port b5)
>> SLB Port B5 # filt enable           (Enable filtering for port b5)
```

9. **Apply and verify the configuration.**

```
>> SLB Port B5 # apply                 (Make your changes active)
>> SLB Port B5 # cur                   (View current settings for port b5)
```

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

10. **Save your new configuration changes.**

```
>> SLB Port B5 # save                  (Save for restore after reboot)
```

11. **Check the server load balancing information.**

```
>> SLB Port B5 # /info/slb/dump        (View SLB information)
```

Check that all SLB 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 do not take effect until the port's session information is updated (every two minutes or so). 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 7.2 Command Reference*).

Network Address Translation Examples

In the following Network Address Translation (NAT) examples, a company has configured its internal network with *private* IP addresses. A private network is one that is isolated from the global Internet and is, therefore, free from the usual restrictions requiring the use of registered, globally unique IP addresses.

Private networks serve two main purposes: First, because private IP addresses are not valid or visible outside the private network, they can increase network security: Second, since valid, registered IP addresses are a limited resource, many companies use private IP addresses to create internal networks much larger than could be created using only their official addresses.

With NAT, private networks are not required to remain isolated. NAT capabilities within the switch allow internal, private network IP addresses to be translated to valid, publicly advertised IP addresses and back again.

Internal Client Access to the Internet

In this dynamic NAT example, clients on the internal private network require TCP/UDP access to the Internet. This is a many-to-one solution: multiple clients on the private subnet take advantage of a single external IP address, thus conserving valid IP addresses:

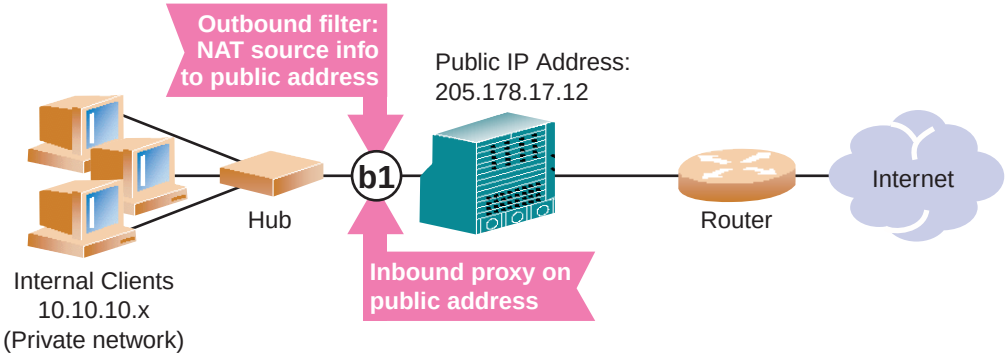


Figure 8-5 Dynamic Network Address Translation

This example requires a NAT filter to be configured on the switch port connected to the internal clients. When the NAT filter is triggered by outbound client traffic, the internal private IP address information on the outbound packets is translated to a valid, publicly advertised IP address. In addition, the public IP address must be configured as a proxy IP address on the switch port connected to the internal clients. The proxy performs the reverse translation, restoring the private network addresses on inbound packets.

This example could be configured as follows:

NOTE – The `invert` option is only specific to this example and is not required.

```
>> # /cfg/slb/filt 14                    (Select the menu for client filter)
>> Filter 14 invert ena                   (Invert the filter logic)
>> Filter 14 dip 10.10.10.0               (If the destination is not private)
>> Filter 14 dmask 255.255.255.0          (For the entire private subnet range)
>> Filter 14 sip any                      (From any source IP address)
>> Filter 14 action nat                   (Perform NAT on matching traffic)
>> Filter 14 nat source                   (Translate source information)
>> Filter 14 ena                          (Enable the filter)
>> Filter 14 adv/proxy enable             (Allow PIP proxy translation)
>> Filter 14 /cfg/slb/port b1             (Select SLB port b1)
>> SLB port B1# add 14                    (Add the filter to port b1)
>> SLB port B1# pip 205.178.17.12         (Set public IP address proxy)
>> SLB port B1# filt enable               (Enable filtering on port b1)
>> SLB port B1# apply                     (Apply configuration changes)
>> SLB port B1# save                      (Save configuration changes)
```

NOTE – Dynamic NAT solutions apply only to TCP/UDP traffic. Also, filters for dynamic NAT should be placed behind any static NAT filters (see Figure 8-6). Dynamic filters should be given higher filter numbers.

External Client Access to a Server

In this example, clients on the external Internet require access to a server on the private network:

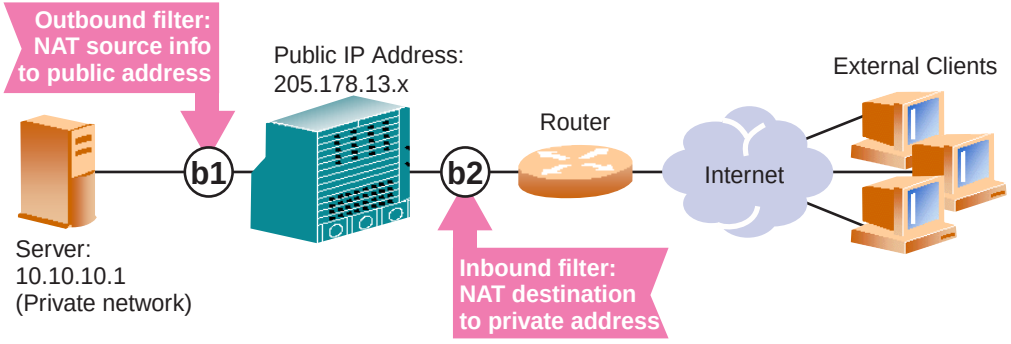


Figure 8-6 Static Network Address Translation

This static NAT (non-proxy) example requires two filters: one for the external client-side switch port, and one for the internal, server-side switch port. The client-side filter translates incoming requests for the publicly advertised server IP address to the server's internal private network address. The filter for the server-side switch port reverses the process, translating the server's private address information to a valid public address.

This could be configured as follows:

```
>> # /cfg/slb/filt 10                    (Select the menu for outbound filter)
>> Filter 10# action nat                  (Perform NAT on matching traffic)
>> Filter 10# nat source                  (Translate source information)
>> Filter 10# sip 10.10.10.0              (From the clients private IP address)
>> Filter 10# smask 255.255.255.0         (For the entire private subnet range)
>> Filter 10# dip 205.178.13.0            (To the public network address)
>> Filter 10# dmask 255.255.255.0         (For the same subnet range)
>> Filter 10# adv/proxy disable           (Override any PIP proxy settings)
>> Filter 10# ena                         (Enable the filter)
>> Filter 10# ../filt 11                  (Select the menu for inbound filter)
>> Filter 11# action nat                  (Use the same settings as outbound)
>> Filter 11# nat dest                    (Reverse the translation direction)
>> Filter 11# sip 10.10.10.0              (Use the same settings as outbound)
>> Filter 11# smask 255.255.255.0         (Use the same settings as outbound)
>> Filter 11# dip 205.178.13.0            (Use the same settings as outbound)
>> Filter 11# dmask 255.255.255.0         (Use the same settings as outbound)
>> Filter 11# adv/proxy disable           (Override any PIP proxy settings)
>> Filter 11# ena                         (Enable the filter)
>> Filter 11# ../port b1                  (Select server-side port)
>> SLB port B1# add 10                    (Add the outbound filter)
>> SLB port B1# filt enable               (Enable filtering on port b1)
>> SLB port B1# ../port b2                (Select the client-side port)
>> SLB port B2# add 11                    (Add the inbound filter)
>> SLB port B2# filt enable               (Enable filtering on port b2)
>> SLB port B2# apply                     (Apply configuration changes)
>> SLB port B2# save                      (Save configuration changes)
```

Note the following important points about this configuration:

- Within each filter, the `smask` and `dmask` values are identical.
- All parameters for both filters are identical except for the NAT direction. For filter 10, `nat source` is used. For filter 11, `nat dest` is used.
- Filters for static (non-proxy) NAT should be placed ahead of dynamic NAT filters. Static filters should be given lower filter numbers.

CHAPTER 9

Application Redirection

Application Redirection improves network bandwidth and provides unique network solutions. Filters can be created to redirect traffic to cache and application servers. 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. The following topics are discussed in this chapter:

- "Overview" on page 146. Application redirection helps reduce the traffic congestion during peak loads by accessing locally cached information. This section also discusses how performance is improved by balancing cached Web requests across multiple servers.
- "Web Cache Configuration Example" on page 148. This section provides a step-by-step procedure on how to intercept all Internet bound HTTP requests (on default TCP port 80), and redirect them to the Web cache servers.
- "IP Proxy Addresses for NAT" on page 153. This section discusses the benefits of transparent proxies when used with application redirection.
- "Excluding Noncacheable Sites" on page 154. This section describes how to filter out applications that keep real-time session information from being redirected to cache servers.

NOTE – To access Application Redirection functionality, the optional Layer 4 software must be enabled on the switch (see Chapter 6 of the *Web OS 7.2 Command Reference*).

Overview

Much of the information downloaded from the Internet is not unique, as clients will often access the same Web page many times for additional information or to explore other links. Duplicate information also gets requested as the components that make up Internet data at a particular Web site (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.

Application redirection can help reduce the traffic congestion during peak loads. When Application redirection filters are properly configured for the Web OS switch, outbound client requests for Internet data are intercepted and redirected to a group of application or Web cache servers on your network. The servers duplicate and store inbound Internet data that has been requested by your clients. If the servers recognize a client's outbound request as one that can be filled with cached information, the servers will supply the information rather than send the request out across the Internet.

In addition to increasing the efficiency of your network, accessing locally cached information can be much faster than 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.

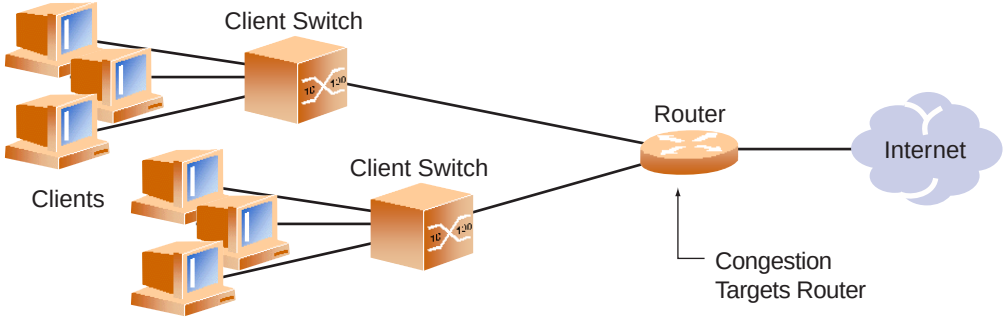


Figure 9-1 Traditional Network Without Web Cache Redirection

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.

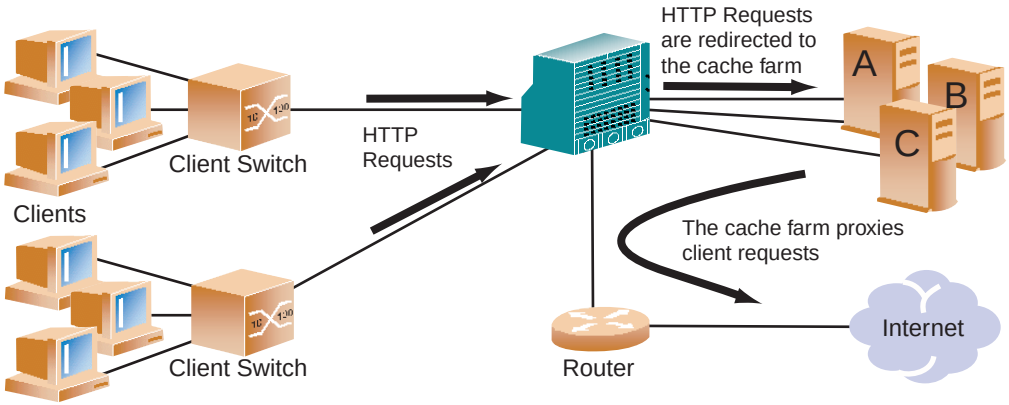


Figure 9-2 Network with Web Cache Redirection

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.

Additional Application Redirection Options

Application redirection can be used in combination with other Layer 4 options, such as load balancing metrics, health checks, real server group backups, and more. See "Additional SLB Options" on page 117 for details.

Web Cache Configuration Example

The following is required prior to configuration:

- You must be connected to the switch Command Line Interface (CLI) 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 7.2 Command Reference*.

In this example, an Alteon 708 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 Command Reference* for a list of other well-known protocols and services.

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

   Similar to server load balancing, the Web cache real servers are 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 NAT" on page 153).

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

   Table 9-1 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 transparent Web cache software on all three Web cache servers.**

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

   Since, by default, the Web switch only remaps 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:

```
>> # /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)
>> IP Interface 1# apply                  (Make your changes active)
```

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. **Define each real server.**

   For each Web cache real server, you must assign a real server number, specify its actual IP address, and enable 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. **Define a real server group.**

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

```
>> Real server 3 # ../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. **Set the real server group metric to `minmisses`.**

   This setting minimizes Web cache misses in the event real servers fail or are taken from service:

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

7. **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 6 of the *Web OS 7.2 Command Reference*.

---

8. **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 Network Address Translation (NAT) and proxy addresses are used on the switch (see Step 2 on page 149), 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 NAT" on page 153 for more information on IP proxy addresses.

9. **Create a default filter.**

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

```
>> Filter 2# ../filt 1024                  (Select the default filter)
>> Filter 1024# sip any                    (From any source IP addresses)
>> Filter 1024# dip any                    (To any destination IP addresses)
>> Filter 1024# proto any                  (For any protocols)
>> Filter 1024# action allow               (Set the action to allow traffic)
>> Filter 1024# enable                     (Enable the default filter)
```

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

10. **Assign the filters to the client ports.**

    Assuming that the redirected clients are connected to physical switch ports a5 and a6, both ports are configured to use the previously created filters as follows:

```
>> Filter 1024# ../port a5                 (Select the SLB port a5)
>> SLB port A5 # add 2                     (Add filter 1 to port a5)
>> SLB port A5 # add 1024                  (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 1024                  (Add the default filter to port a6)
>> SLB port A6 # filt enable               (Enable filtering for port a6)
```

11. **Enable, apply, and verify the configuration.**

```
>> SLB port A6 # ../cfg/slb                (Select Server Load Balancing Menu)
>> Layer 4 # on                            (Activate Layer 4 software services)
>> Layer 4 # apply                         (Make your changes active)
>> Layer 4 # 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 7.2 Command Reference*.

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

13. **Save your new configuration changes.**

```
>> Layer 4 # save                    (Save for restore after reboot)
```

14. **Check the server load balancing information.**

```
>> Layer 4 # /info/slb/dump          (View all SLB information)
```

Check that all SLB 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 affect 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 7.2 Command Reference*).

IP Proxy Addresses for NAT

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 proxy IP addresses configured on redirected ports, the Web switch can redirect client requests to servers located on any subnet, anywhere.
- The Web 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# ../port a6                (Select network port a6)
>> SLB port A6# pip 200.200.200.69        (Set proxy IP address for 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 this example) with the real server TCP or UDP port to which redirected traffic is sent. In this case, the requests are mapped to a 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 Table 8-2 on page 131.

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 number>/filter <filter number>
```

Excluding Noncacheable Sites

Some Web sites provide content that is not well suited for redirection to cache servers. Such sites might provide browser-based games or 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 want to prevent a popular Web-based game site on subnet 200.10.10.0 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)
```

CHAPTER 10

Health Checking

Content intelligent Web switches allow Webmasters to customize server health checks to verify content accessibility in large Web sites. As the amount of content grows and information is distributed across different server farms, flexible, customizable content health checks are critical to ensure end-to-end availability.

Alteon 708 switches monitor the health of servers and applications using the health-checking mechanisms described in the following sections in this chapter:

- "Real Server Health Check Parameters" on page 156. This section explains the switch's default health check which checks the status of each service on each real server every two seconds.
- "Health Check Types" on page 157. This section describes the different types of health checking: TCP and application-specific health checks, ASCII script-based health checks, and ICMP health check.
- "Hostname for HTTP Content Health Checks" on page 158. This section provides examples of HTTP-based health checks using hostnames.
- "RADIUS Server Health Checks" on page 160. This section explains how the Remote Authentication Dial-In User Service (RADIUS) protocol is used to authenticate dial-up users to Remote Access Servers (RASs).
- "IMAP Server Health Checking" on page 161. This section describes how the mail server protocol is used to perform health checking between a client system and a mail server.
- "Script-Based Health Checks" on page 161. This section describes how to configure the switch to send a series of health-check requests to real servers or real server groups and monitor the responses.
- "HTTPS/SSL Health Check" on page 166. This section explains how the switch queries the health of the SSL servers by sending an SSL client "Hello" packet and then verifies the content of the server's "Hello" response.
- "Failure Types" on page 167. This section explains the *Service Failed* and *Server Failed* states.

Real Server Health Check Parameters

Alteon 708s running server load balancing monitor the servers in the real server group and the load-balanced application(s) running on them. If a session switch detects that a server or application has failed, it will not direct any new connection requests to that server. When a service fails, an Alteon Web switch can remove the individual service from the load-balancing algorithm without affecting other services provided by that server.

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. If a service does not respond to four consecutive health checks, the switch, by default, declares the service unavailable. You can modify both the health check interval and the number of retries.

`>> # /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)*

NOTE – Health checks are performed sequentially when used in conjunction with a virtual server configured with multiple services and groups. As a result, the actual health check interval could vary significantly from the value set for it using the `inter` parameter.

Health Check Types

Layer 4 health check support includes TCP and application-specific health checks and ASCII script-based health checks. Layer 3 health checking is available for ICMP. Each health check type is summarized below:

TCP Health Checks

TCP health checks are useful in verifying TCP applications that cannot be scripted. Session switches monitor the health of servers and applications by sending Layer 4 connection requests (TCP SYN packets) for each load-balanced TCP service to each server in the server group on a regular basis. The rate at which these connection requests are sent is a user-configurable parameter. These connection requests identify both failed servers and failed services on a healthy server. When a connection request succeeds, the session switch quickly closes the connection by sending a TCP FIN packet.

Application-Specific Health Checks

These health checks include the following applications:

- HTTP
- FTP
- SMTP
- POP3
- IMAP
- NNTP
- RADIUS

  The Web switch can check availability of Remote Access Dial-In User Service (RADIUS) content and SSL connections.

ICMP Health Checks

The Layer 3 echo/echo reply (ping) health check is used for UDP services or when ICMP health checks are configured.

Script-Based Health Checks

The “send/expect” script-based health checks dynamically verify application and content availability using scripts. These scripts execute a sequence of tests to verify application and content availability. For more information, see “Script-Based Health Checks” on page 161.

Hostname for HTTP Content Health Checks

HTTP-based health checks can include the hostname for `HOST:` headers. The `HOST:` header and health-check URL are constructed from the following components:

Table 10-1 HTTP Content Health Check Components

Item	Option	Configured Under	Maximum Length
Virtual server hostname	`hname`	`/cfg/slb/virt/service`	9 characters
Domain name	`dname`	`/cfg/slb/virt`	35 characters
Server group health check field	`content`	`/cfg/slb/group`	34 characters

If the `HOST:` header is required, an `HTTP/1.1 GET` will occur; otherwise an `HTTP/1.0 GET` will occur.

Example 1:

```
hname   = everest
dname   = alteonwebsystems.com
content = index.html
```

Health check is performed using:

```
GET /index.html HTTP/1.1
Host: everest.alteonwebsystems.com
```

Example 2:

```
hname   = (none)
dname   = raleighduram.cityguru.com
content = /page/gen/?_template=alteon
```

Health check is performed using:

```
GET /page/gen/?_template=alteon HTTP/1.1
Host: raleighduram.cityguru.com
```

Example 3:

```
hname   = (none)
dname   = compute
content = index.html
```

Health check is performed using:

```
GET /index.html HTTP/1.1
Host: compute
```

Example 4:

```
hname   = (none)
dname   = (none)
content = index.html
```

Health check is performed using:

`GET /index.html HTTP/1.0` (since no `HTTP HOST:` header is required)

Example 5:

```
hname   = (none)
dname   = (none)
content = //everest/index.html
```

Health check is performed using:

```
GET /index.html HTTP/1.1
Host: everest
```

RADIUS Server Health Checks

The Remote Authentication Dial-In User Service (RADIUS) protocol is used to authenticate dial-up users to Remote Access Servers (RASs) and the client application they will use during the dial-up connection.

- RADIUS Content Health Check Enhancements
  - Include the switch IP as the NAS IP parameter in the RADIUS content health check

    The switch uses the IP address of the IP interface that is on the same subnet of the RADIUS server or the default gateway as the NAS IP.

NOTE – Although the NAS IP is not specified as being required in the Request for Comments (RFC), Lucent's native RADIUS servers require this parameter.

  - RADIUS health check using the configured real server port, `rport`

    The RADIUS health check will be performed using the configured real server port (`rport`). To configure RADIUS health checks, use the `/cfg/slb/virt` *number*`/service` *port number* menu.

  - Variable-length RADIUS secret password. Supports less than 16 octets and up to 32 octets

  RADIUS is stateless and uses UDP as its transport protocol. To support RADIUS health checking, the network administrator must configure two parameters in the switch: the `/cfg/slb/secret` value and the `content` parameter with a *username:password* value.

```
>> # /cfg/slb/group <real server group number>     (Select the real server group)
>> # health radius                                 (Specify the type of health checking)
>> # content <username>:<password>                 (Specify the RADIUS security values)
>> # /cfg/slb/adv/secret <RADIUS coded value>      (Enter up to 32 characters for
                                                    encrypting/decrypting password)
```

- The `secret` value is a field of up to 32 alphanumeric characters used by the switch to encrypt a password during the RSA Message Digest Algorithm (MD5) and by the RADIUS server to decrypt the password during verification.
- The `content` option specifies the *username:password* value that the server tries to match in its user database. In addition to verifying the user name and password, the database may specify the client(s) or port(s) the user is allowed to access.

IMAP Server Health Checking

Internet Message Access Protocol (IMAP) is a mail server protocol used between a client system and a mail server that allows a user to retrieve and manipulate mail messages. IMAP is not used for mail transfers between mail servers.

IMAP servers listen to TCP port 143. To support IMAP health checking, the network administrator must configure a *username:password* value in the switch, using the `content` option on the SLB Real Server Group Menu (`/cfg/slb/group`).

```
>> # /cfg/slb/group real server group number     (Select the real server group)
>> # slb/group/health imap                       (Specify the type of health checking)
>> # slb/group/content username:password         (Specify the IMAP security values)
```

The `content` option specifies the *username:password* value that the server tries to match in its user database. In addition to verifying the user name and password, the database may specify the client(s) or port(s) the user is allowed to access.

Script-Based Health Checks

Using this feature, you can configure the switch to send a series of health check requests to real servers or real server groups and monitor the responses. ASCII-based scripts can be used to verify application and content availability.

NOTE – Only TCP services can be health checked, since UDP protocols are usually not ASCII based.

The benefits of using script-based health checks are listed below:

- Ability to send multiple commands
- Check for any return string
- Test availability of different applications
- Test availability of multiple domains or Web sites

Web OS supports the following capacity for a single switch:

- 1024 bytes per script
- 8 scripts per switch
- approximately 10 to 15 health-check statements (HTTP GET and expect strings)

A simple command line interface controls the addition and deletion of ASCII commands to each script. New commands are added and removed from the end of the script. Commands exist to open a connection to a specific TCP port, send an ASCII request to the server, expect an ASCII string, and close a connection. The string configured with an `expect` command is searched for in each response packet. If it is not seen in any response packet before the real server health check interval expires, the server does not pass the expect step and fails the health check. A script can contain any number of these commands, up to the allowable number of characters that a script supports.

NOTE – Health check scripts can only be set up via the Command Line Interface (CLI), but once entered, can be assigned as the health check method using SNMP or the Browser-Based Interface (BBI).

Script Format

The general format for health check scripts is shown below:

```
open <application port>                    (e.g., 80 for HTTP, 23 for Telnet, etc.)
send <request1>
expect <response1>
send <request2>
expect <response2>
send <request3>
expect <response3>
close
```

NOTE – For HTTP 1.1 pipelining, you must open and close each response in the script individually.

Consider the following factors when creating a health check script:

- Each script should start with the command `open` <*protocol port number*>. The next line can be either a `send` or `expect`.
- The first word is the method. This is usually `GET`; however, HTTP supports several other commands, including `PUT` and `HEAD`. The second word indicates the content desired, or request URI, and the third word represents the version of the protocol used by the client.

  If you supplied HTTP/1.1 for the protocol version, you would also have to add in the following line: `Host: www.hostname.com`

  Example: `GET /index.html HTTP/1.1` (press Enter key)
  `Host: www.hostname.com` (press Enter key twice)

  This is known as a host header. It is important to include because most Web sites now require it for proper processing. Host headers were optional in HTTP/1.0 but are required when you use HTTP/1.1+.
- In order to tell the Web server you have finished entering header information, a blank line of input is needed after all headers. At this point, the URL will be processed and the results returned to you.

NOTE – If you make an error, type **`rem`** to remove the last typed script line entered. If you need to remove more than one line, type **`rem`** for the each line that needs to be removed.

- The switch provides the "\" prompt, which is one enter key stroke. When using the send command, note what happens when you type the send command with the command string. When you type send, press enter and allow the switch to format the command string (that is, \ versus \\).

Scripting Guidelines

- Use generic result codes that are standard and defined by the RFCs that govern the application. This helps ensure that if the customer changes server software, the servers won't start failing unexpectedly.
- Search only for the smallest and most concise piece of information possible. Each script cannot exceed 1K in size, so use the space wisely.
- Avoid tasks that may take a long time to perform or the health check will fail. For example, avoid tasks that exceed the interval for load balancing.

Script Configuration Example

Configure the switch to check a series of Web pages (HTML or dynamic CGI scripts) before it declares a real server is available to receive requests.

NOTE – If you are using the CLI to create a health check script, you must use quotes (“) to indicate the beginning and end of each command string.

```
/cfg/slb/group x/health script1/content none
/cfg/slb/adv/script1

open 80
send "GET /index.html HTTP/1.1\\r\\nHOST:www.hostname.com\\r\\n\\r\\n"
expect "HTTP/1.1 200"
close
open 80
send "GET /script.cgi HTTP/1.1\\r\\nHOST:www.hostname.com\\r\\n\\r\\n"
expect "HTTP/1.1 200"
close
open 443
…
close
```

NOTE – When you are using the CLI to enter the send string as an argument to the `send` command, you must type two “\”s before an “n” or “r.” If you are instead prompted for the line, that is, the text string is entered after hitting <return>, then only one “\” is needed before the “n” or “r.”

A sample script is shown below:

```
/cfg/slb/group x/health script2/content none
/cfg/slb/adv/script2

open 80
send "GET /images/default.asp HTTP/1.1\\r\\nHOST: 192.192.1.2\\r\\n\\r\\n"
expect "HTTP/1.1 200"
close

open 80
send "GET /install/default.html HTTP/1.1\\r\\nHOST: 192.192.1.2\\r\\n\\r\\n"
expect "HTTP/1.1 200"
close

open 80
send "GET /script.cgi HTTP/1.1\\r\\nHOST: www.alteon.com\\r\\n\\r\\n"
expect "HTTP/1.1 200"
close
```

Script-based health checking is intelligent in that it will only send the appropriate requests to the relevant servers. In the example above, the first GET statement will only be sent to RIP 1 and RIP 2. Going through the health check statements serially will ensure that all content is available by at least one real server on the remote site.

Configure the remote real server IP address (the virtual server IP address of the remote site) to accept `any` URL requests. The purpose of the first GET is to check if RIP 1 or RIP 2 is up—that is, to check if the remote site has at least one server for “images” content. Either RIP 1 or RIP 2 will respond to the first GET health check.

If all the real server IP addresses are down, RIP 7 (the virtual server IP address of the remote site) will respond with an HTTP Redirect (respond code 302) to the health check. Thus, the health check will fail as the expected respond code is 200, ensuring that the HTTP Redirect messages will not cause a loop.

Verifying Script-Based Health Checks

If a script fails, the expect line in the script that is not succeeding is displayed under the `/info/slb/real` <*real server number*> `command`:

```
>> # /info/slb/real 1
  1: 205.178.13.225, 00:00:00:00:00:00, vlan 1, port 0, health 4, FAILED
    real ports:
    script 2, DOWN, current
      send GET / HTTP/1.0\r\n\r\n
      expect HTTP/1.0 200
```

The server is not responding to the `GET` with the expect string.

When the script succeeds in determining the health of a real server, the following information is displayed:

```
>> # /info/slb/real 1
  1: 205.178.13.223, 00:00:5e:00:01:24, vlan 1, port 2, health 4, up
    real ports:
    script 2, up, current
```

HTTPS/SSL Health Check

The `sslh` health check option on the Real Server Group Menu (`/cfg/slb/group/healt/sslh`) allows the switch to query the health of the SSL servers by sending an SSL client “Hello” packet and then verify the contents of the server’s “Hello” response. SSL health check is performed using the real server port configured, that is, the `rport`.

The SSL enhanced health check behavior is summarized below:

- The switch sends a SSL “Hello” packet to the SSL server.
- If it is up and running, the SSL server responds with the “Server Hello” message.
- The switch verifies various fields in the response and marks the service “UP” if the fields are OK.

During the handshake, the user and server exchange security certificates, negotiate an encryption and compression method, and establish a session ID for each session.

Failure Types

Service Failure

If a certain number of connection requests for a particular service fail, the session switch places the service into the “service failed” state. While in this state, no new connection requests are sent to the server for this service; however, if graceful real server failure is enabled (`/cfg/slb/adv/grace ena`), state information about existing sessions is maintained and traffic associated with existing sessions continues to be sent to the server. Connection requests to, and traffic associated with, other load-balanced services continue to be processed by the server.

Example: A real server is configured to support HTTP and FTP within two real server groups. If a session switch detects an HTTP service failure on the real server, it removes that real server group from the load-balancing algorithm for HTTP but keeps the real server in the mix for FTP. Removing only the failed service from load balancing allows users access to all healthy servers supporting a given service.

When a service on a server is in the “service failed” state, the session switch sends Layer 4 connection requests for the failed service to the server. When the session switch has successfully established a connection to the failed service, the service is restored to the load-balancing algorithm.

Server Failure

If all load-balanced services supported on a server fail to respond to switch connection requests within the specified number of attempts, then the server is placed in the “server failed” state. While in this state, no new connection requests are sent to the server; however, if graceful real server failure is enabled (`/cfg/slb/adv/grace ena`), state information about existing sessions is maintained and traffic associated with existing sessions continues to be sent to the server.

NOTE – All load-balanced services on a server must fail before the switch places the server in the *server failed* state.

The server is brought back into service as soon as the first service is proven to be healthy. Additional services are brought online as they are subsequently proven to be healthy.

CHAPTER 11
High Availability

Alteon Web switches support high-availability network topologies through an enhanced implementation of the Virtual Router Redundancy Protocol (VRRP).

In a high-availability network topology, no device can create a single point-of-failure for the network or force a single point-of-failure to any other part of the network. This means that your network will remain in service despite the failure of any single device. To achieve this usually requires redundancy for all vital network components.

The following topics are discussed in this chapter:

- "VRRP Overview" on page 170. This section discusses VRRP operation and Alteon WebSystems' redundancy configurations.
- "Failover Methods" on page 175. This section describes the three modes of high availability.
- "Web OS Extensions to VRRP" on page 181. This section describes VRRP enhancements implemented in Web OS.
- "High Availability Configurations" on page 184. This section discusses a few of the more useful and easily deployed redundant configurations.
  - Active-Standby Virtual Server Router Configuration
  - Active-Active VIR and VSR Configuration
  - Active/Active Server Load Balancing Configuration
  - VRRP-Based Hot-Standby Configuration
- "Virtual Router Deployment Considerations" on page 199. This section describes issues to consider when deploying virtual routers.

VRRP Overview

To understand the operation of Alteon WebSystems' redundancy configurations, this section describes VRRP operation and the Alteon-specific extensions to VRRP.

VRRP enables redundant router configurations within a LAN, providing alternate router paths for a host to eliminate single points-of-failure within a network. Each participating VRRP-capable routing device is configured with the same virtual router IP address and ID number. One of the virtual routers is elected as the master, based on a number of priority criteria, and assumes control of the shared virtual router IP address. If the master fails, one of the backup virtual routers will take control of the virtual router IP address and actively process traffic addressed to it.

Because the router associated with a given alternate path supported by VRRP uses the same IP address and MAC address as the routers for other paths, the host's gateway information does not change, no matter what path is used. A VRRP-based redundancy schema reduces administrative overhead because hosts need not be configured with multiple default gateways.

VRRP Components

Each physical router running VRRP is known as a *VRRP router*.

Virtual Interface Router

Two or more VRRP routers can be configured to form a *virtual interface router (VIR)*. (RFC 2338 calls this entity a "virtual router.") The term virtual interface router will be used to distinguish this type of entity from a *virtual server router (VSR)*, as described in "Web OS Extensions to VRRP" on page 181. When the term *virtual router* is used herein, the concept applies to both virtual interface routers and virtual server routers. Each VRRP router may participate in one or more virtual interface routers.

A virtual interface router acts as a default or next hop gateway for hosts on a LAN. Each virtual interface router consists of a user-configured *virtual router identifier* (VRID) and an IP address.

Virtual Router MAC Address

The VRID is used to build the *virtual router MAC Address*. The five highest-order octets of the virtual router MAC Address are the standard MAC prefix (00-00-5E-00-01) defined in RFC 2338. The VRID is used to form the lowest-order octet.

Owners and Renters

Only one of the VRRP routers in a virtual interface router may be configured as the IP address owner. This router has the virtual interface router's IP address as its real interface address. This router responds to packets addressed to the virtual interface router's IP address for ICMP pings, TCP connections, and so on.

There is no requirement for any VRRP router to be the IP address owner. Most VRRP installations choose not to implement an IP address owner. For the purposes of this chapter, VRRP routers that are not the IP address owner are called *renters*.

Master and Backup Virtual Router

Within each virtual router, one of the VRRP routers is selected to be the virtual router master. See "Selecting the Master VRRP Router" on page 173 for an explanation of the selection process.

NOTE – If the IP address owner is available, it will always become the virtual router master.

The virtual router master forwards packets sent to the virtual interface router. It also responds to Address Resolution Protocol (ARP) requests sent to the virtual interface router's IP address. Finally, the virtual router master sends out periodic advertisements to let other VRRP routers know it is alive and its priority (explained below).

Within a virtual router, the VRRP routers not selected to be the master are known as virtual router backups. Should the virtual router master fail, one of the virtual router backups becomes the master and assumes its responsibilities.

The Alteon Web switches in Figure 11-1 have been configured as VRRP routers. Together, they form a *virtual interface router (VIR)*.

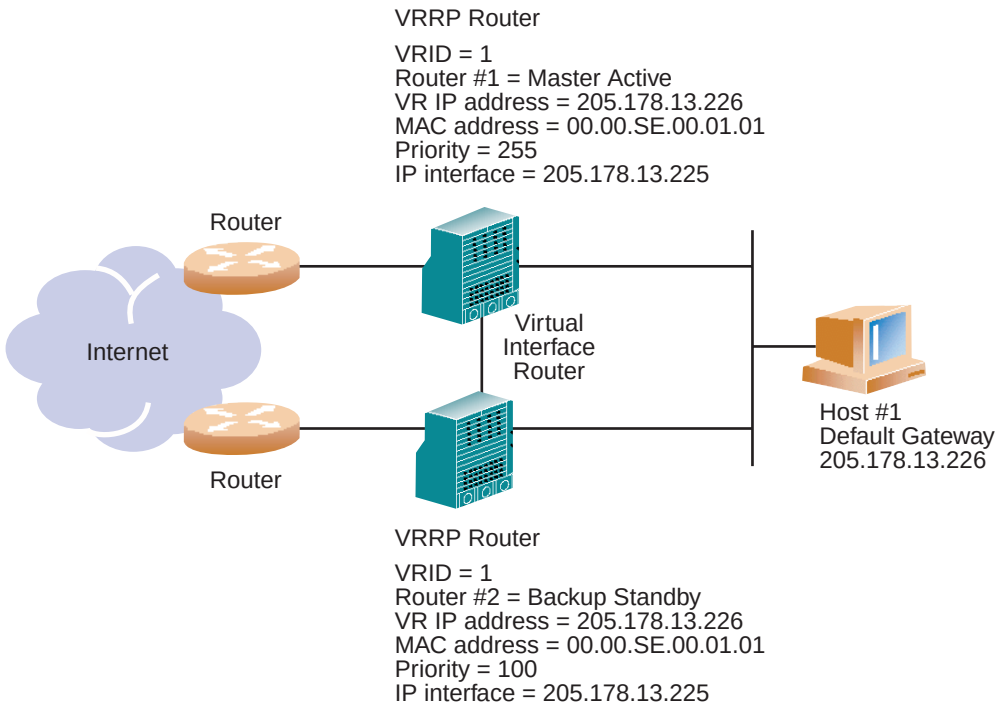


Figure 11-1 Example 1: VRRP Router

The switch on the left in Figure 11-1 has its real interface configured with the IP address of the virtual interface router and is, therefore, the IP address owner. It also becomes the virtual router master. The switch on the right is a virtual router backup. Its real interface is configured with an IP address that is on the same subnet as the virtual interface router but is not the IP address of the virtual interface router.

The virtual interface router has been assigned a VRID = 1. Therefore, both of the VRRP routers have a MAC address = 00-00-5E-00-01-01.

VRRP Operation

The host shown in Figure 11-1 is configured with the virtual interface router's IP address as its default gateway. The master forwards packets destined to remote subnets and responds to ARP requests. In this example, the master is also the virtual interface router's IP address owner—therefore it also responds to ICMP ping requests and IP datagrams destined for the virtual interface router's IP address. The backup does not forward any traffic on behalf of the virtual interface router nor does it respond to ARP requests.

If the owner is not available, the backup becomes the master and takes over responsibility for packet forwarding and responding to ARP requests. However, because this switch is not the owner, it does not have a real interface configured with the virtual interface router's IP address.

Selecting the Master VRRP Router

Each VRRP router that is not an owner is configured with a priority between 1–254. According to the VRRP standard, an owner has a priority of 255. A bidding process determines which VRRP router is or becomes the master—the VRRP router with the highest priority. Owners have a higher priority than the range permitted for non-owners. If there is an IP address owner, it is always the master for the virtual interface router, as long as it is available.

The master periodically sends out advertisements to an IP multicast address. As long as the backups receive these advertisements, they remain in the backup state. If a backup does not receive an advertisement for three advertisement intervals, it initiates a bidding process to determine which VRRP router has the highest priority and takes over as master.

If, at any time, a backup determines that it has higher priority than the current master does, it can preempt the master and become the master itself, unless configured not to do so. In pre-emption, the backup assumes the role of master and begins to send its own advertisements. The current master sees that the backup has higher priority and will stop functioning as the master.

A backup router can stop receiving advertisements for one of two reasons—the master can be down, or all communications links between the master and the backup can be down. If the master has failed, it is clearly desirable for the backup (or one of the backups, if there is more than one) to become the master.

NOTE – If the master is healthy but communication between the master and the backup has failed, there will be then two masters within the virtual router. To prevent this from happening, configure redundant links be used between the switches that form a virtual router.

Active-Standby Failover

The previous text described the use of a group of VRRP routers to form a single virtual interface router. It implements a traditional *hot-standby* configuration in which the backup router only functions when the active router has failed. VRRP can also be used to implement *active-standby* configurations. In an active-standby configuration, both switches support *active* traffic, but are configured so that they do not simultaneously support the same service.

In the example shown in Figure 11-2, the switch on the left is the master for the virtual interface router with VRID = 1, and it's backup for the virtual interface router with VRID = 2. The switch on the right is master for the virtual interface router with VRID = 2 and backup for the virtual interface router with VRID = 1. In this manner, both routers can actively forward traffic at the same time but not for the same interface.

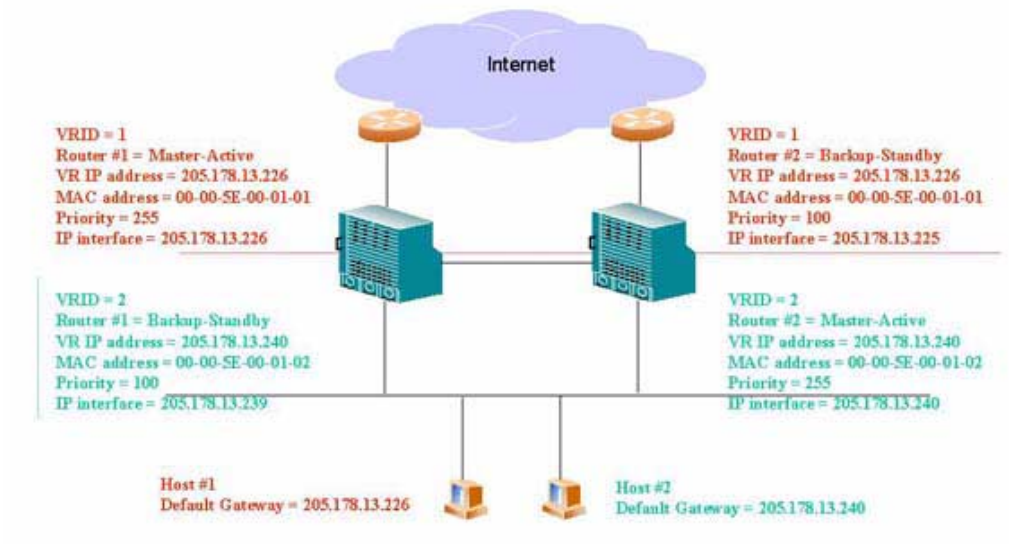


Figure 11-2 Example 2: VRRP Router

Failover Methods

With service availability becoming a major concern on the Internet, service providers are increasingly deploying Internet traffic control devices, such as Web switches, in redundant configurations. Traditionally, these configurations have been *hot-standby* configurations, where one switch is active and the other is in a standby mode. A non-VRRP hot-standby configuration is shown in the figure below:

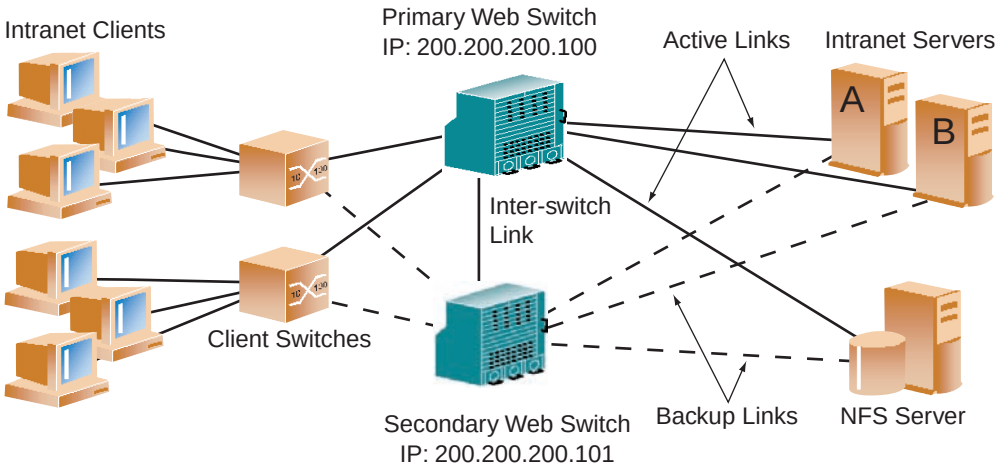


Figure 11-3 A Non-VRRP, Hot-Standby Configuration

While hot-standby configurations increase site availability by removing single points-of-failure, service providers increasingly view them as an inefficient use of network resources because one functional Web switch sits by idly until a failure calls it into action. Service providers now demand that vendors' equipment support redundant configurations where all devices can process traffic when they are healthy, increasing site throughput and decreasing user response times when no device has failed.

Alteon WebSystems high availability configurations are based on the extensions to VRRP that it has developed to support Layer 4 switching services, such as Server Load Balancing (SLB), to support active operation of interfaces (at Layer 3) and services (at Layer 4) across multiple switches at the same time and to interact with Spanning Tree to control frame path.

The Web OS implementation of VRRP supports three modes of high availability:

- Active-Standby
- Active-Active
- Hot-Standby

The first mode, *active-standby*, is based on standard VRRP, as defined in RFC 2338. The second and third modes, *active-active* and *hot-standby*, are based on proprietary Alteon WebSystems extensions to VRRP. Each mode is described in detail in the following sections.

Active-Standby Redundancy

In an active-standby configuration, shown in Figure 11-4, two Web switches are used. Both switches support active traffic but are configured so that they do not simultaneously support the same service. Each switch is active for its own set of services, such as IP routing interfaces or load-balancing virtual IP addresses, and acts as a standby for other services on the other switch. If either switch fails, the remaining switch takes over processing for all services. The backup switch may forward Layer 2 and Layer 3 traffic, as appropriate.

NOTE – In an active-standby configuration, the same service cannot be active simultaneously on both switches.

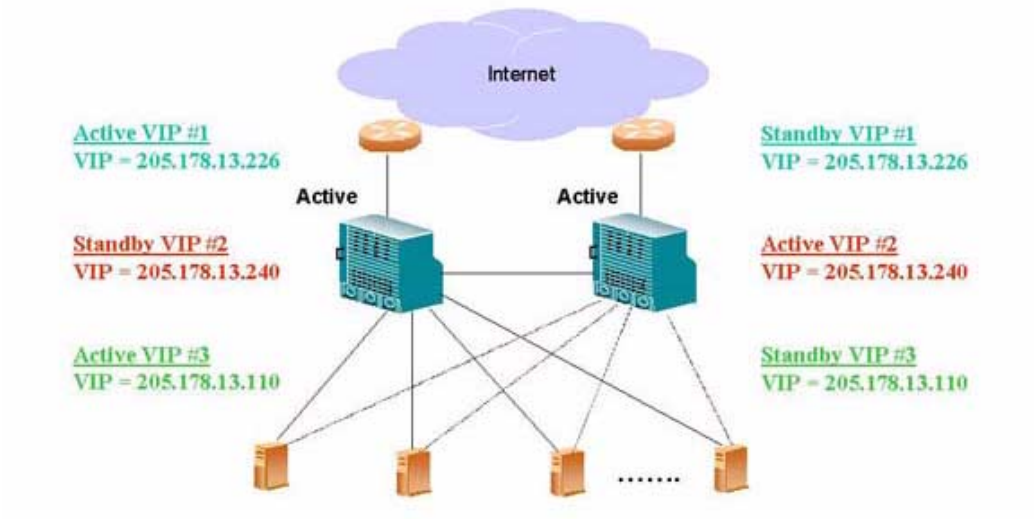


Figure 11-4 Active-Standby Redundancy

Active-Active Redundancy

In an active-active configuration, two Web switches provide redundancy for each other, with both active at the same time for the same services.

Alteon WebSystems has extended VRRP to include virtual servers, allowing full active/active redundancy between its Layer 4 switches. In an active-active configuration, shown in Figure 11-5, both switches can process traffic for the same service at the same time; both switches can be active simultaneously for a given IP routing interface or load-balancing virtual servers.

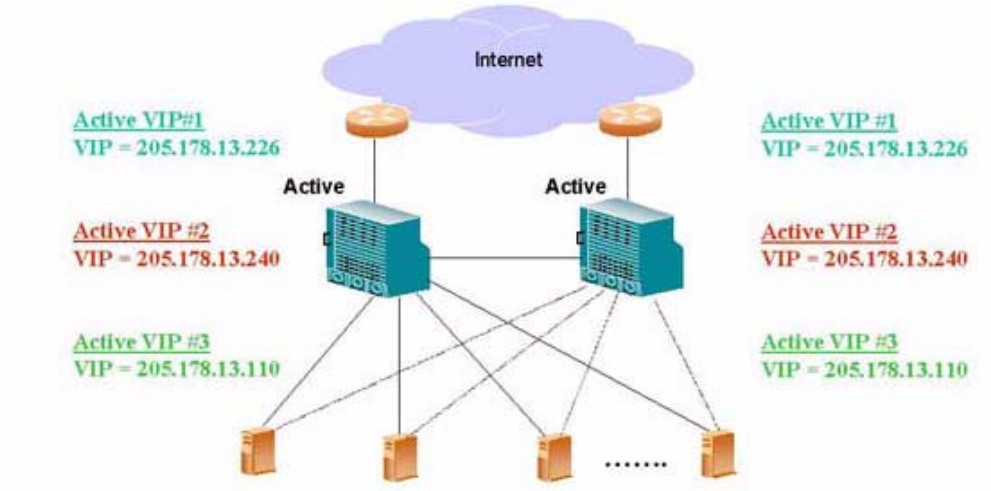


Figure 11-5 Active-Active Redundancy

In the example above, one switch is still the master router. However, traffic going through the backup router (associated with the same virtual router on the switch) that is addressed to the master router will be intercepted and processed by the backup router.

Hot-Standby Redundancy

In a hot-standby configuration, Spanning Tree Protocol (STP) is not needed to eliminate bridge loops. This speeds up failover when a switch fails. The standby switch blocks all ports configured as standby ports, whereas the master switch enables these same ports. Consequently, on a given switch, all virtual routers are either master or backup; they cannot change state individually.

To provide as much flexibility as possible, the old hot-standby approach has been modified to eliminate the problems previously associated with it and is now based on VRRP. In a hot-standby configuration, two or more switches provide redundancy for each other. One switch is

elected *master* and actively processes Layer 4 traffic. The other switches (the backups) assume the master role should the master fail. The backups may forward Layer 2 and Layer 3 traffic as appropriate.

There are three components to the VRRP-based, hot-standby model: the virtual router group, additional Layer 4 port states, and configuration synchronization options. The hot-standby model is shown in Figure 11-6.

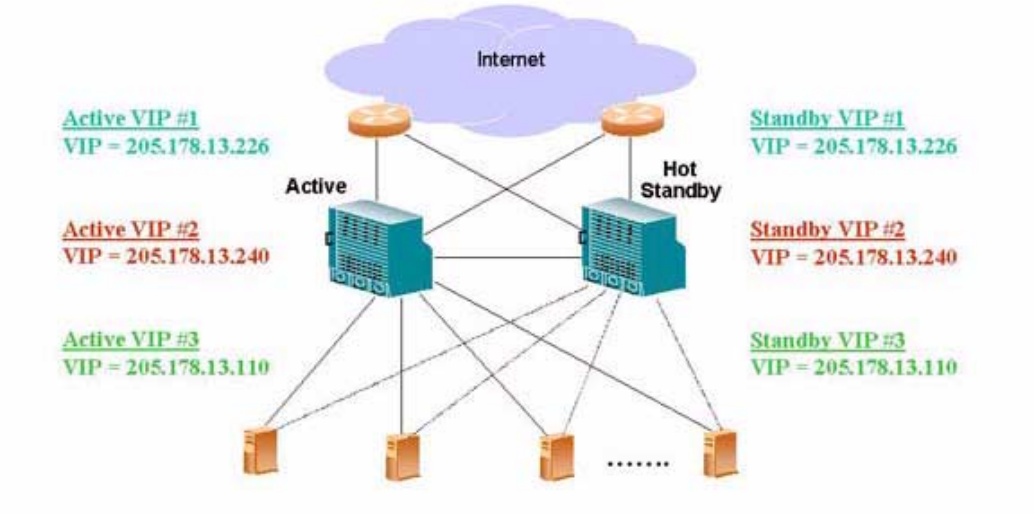


Figure 11-6 Hot-Standby Redundancy

Virtual Router Group

The virtual router group ties all of the virtual routers together as a single entity and is central to the hot-standby configuration. All virtual routers on a given switch must all be either master or backup. They cannot failover individually, only as a group. Once hot-standby is globally enabled, the virtual router group must be enabled. The virtual router group aggregates all of the virtual routers as a single entity.

If the virtual router group is master on one switch, it means the switch is master; otherwise, the switch is backup. However, Layer 4 processing is still enabled. If a virtual server is not a virtual router, the backup switch can still process traffic addressed to that virtual IP address. Filtering is also still functional. Only traffic addressed to virtual server routers is not processed.

VRRP actually contains support for virtual router groups. Each advertisement is not limited to a single virtual router IP address and can include up to 256 addresses. This means that all virtual routers are advertised in the same packet, conserving processing and buffering resources.

However, the advertisements are also used to help bridges learn the virtual router MAC address. Since all of the virtual routers can have different virtual router identifiers (VRIDs), you must rotate the MAC source address of the advertisement to ensure that the bridges learn all of the virtual router MAC addresses.

Hot-Standby and Inter-Switch Port States

The second part of the solution involves introducing two additional Layer 4 port states—hot-standby and inter-switch:

- Links that attach to the standby switch must be configured using the command, **`/cfg/slb/port x/hotstan`**.
- Links that are used by VRRP to deliver updates are configured as `intersw`, or inter-switch link (not to be confused with Cisco's ISL). The command to configure one or more ports as interswitch links is **`/cfg/slb/port`** <*port number*> **`/intersw`**.

NOTE – A port cannot be configured to support both hot-standby and interswitch link.

The hot-standby switch listens to the master's VRRP updates. After an interval period has expired without receiving a update, the backup switch will take over. The forwarding states of hot-standby ports are controlled much like the forwarding states of the old hot-standby approach. Enabling hot-standby on a switch port allows the hot-standby algorithm to control the forwarding state of the port. If a switch is master, the forwarding states of the hot-standby ports are enabled. If a switch is backup, the hot-standby ports are blocked from forwarding or receiving traffic.

When the `hotstan` option (`/cfg/slb/port x/hotstan`) is enabled and all hot-standby ports have link, the virtual router group's priority is automatically incremented by the "track other virtual routers" value. This action allows the switches to failover when a hot-standby port loses link. Other enabled tracking features only have affect when all hot-standby ports on a switch have link. The default virtual routers tracking value is two seconds. Keep in mind that this is an automatic process that cannot be turned off.

NOTE – The VRRP hot-standby approach does *not* support single-link failover. If one hot-standby port loses link, the entire switch must become master to eliminate loss of connectivity.

The forwarding states of non-hot-standby ports are not controlled via the hot-standby algorithm, allowing the additional ports on the switches to provide added port density. The client ports on both switches should be able to process or forward traffic to the master switch.

The inter-switch port state is only a place holder. Its presence forces the user to configure a inter-switch link when hot-standby is globally enabled and prohibits the inter-switch link from also being a hot-standby link for VRRP advertisements. These advertisements must be able to reach the backup switch.

Synchronizing Configurations

The final piece in configuring a high-availability solution includes the addition of synchronization options to simplify the manual configuration. Configuration options have been added to refine what is synchronized, to whom, and to disable synchronizing certain configurations. These options include proxy IP addresses, Layer 4 port configuration, filter configuration, and virtual router priorities.

Also, a peer menu (`cfg/slb/sync/peer`) has been added to allow the user to configure the IP addresses of the switches that should be synchronized. This provides added synchronization validation but does not require the users to enter the IP address of the redundant switch for each synchronization.

NOTE – When using both VRRP and GSLB, you must change the `/cfg/sys/wport` (Browser-Based Interface port) value of the target switch (the switch that is being synchronized to) to a port other than port 80 before VRRP synchronization begins.

For more information on synchronizing configurations between two switches, see "Synchronizing Configurations" on page 203.

Web OS Extensions to VRRP

This section describes the following VRRP enhancements that are implemented in Web OS:

- Virtual Server Routers
- Sharing/Active-Active Failover
- Tracking VRRP Router Priority

Virtual Server Routers

Web OS supports *virtual server routers*, which extend the benefits of VRRP to virtual server IP addresses that are used to perform SLB.

Virtual server routers operate for virtual server IP addresses in much the same manner as Virtual Interface Routers (VIRs) operate for IP interfaces. A master is negotiated via a bidding process, during which information about each VRRP router's priority is exchanged. Only the master can process packets that are destined for the virtual server IP address and respond to ARP requests.

One difference between *virtual server routers* and *virtual interface routers* is that a virtual server router cannot be an *IP address owner.* All virtual server routers are *renters*.

All virtual routers, whether virtual server routers or virtual interface routers, operate independently of one another; that is, their priority assignments, advertisements, and master negotiations are separate. For example, when you configure a VRRP router's priority in a virtual server router, you are not affecting that VRRP router's priority in any virtual interface router or any other virtual server router of which it is a part. However, because of the requirement that MAC addresses be unique on a LAN, VRIDs must be unique among all virtual routers, whether virtual interface routers or virtual server routers.

Sharing/Active-Active Failover

Web OS supports *sharing* of interfaces at both Layer 3 and Layer 4, as shown in Figure 11-7. With sharing, an IP interface or a virtual server IP address can be active simultaneously on multiple switches, enabling active-active operation.

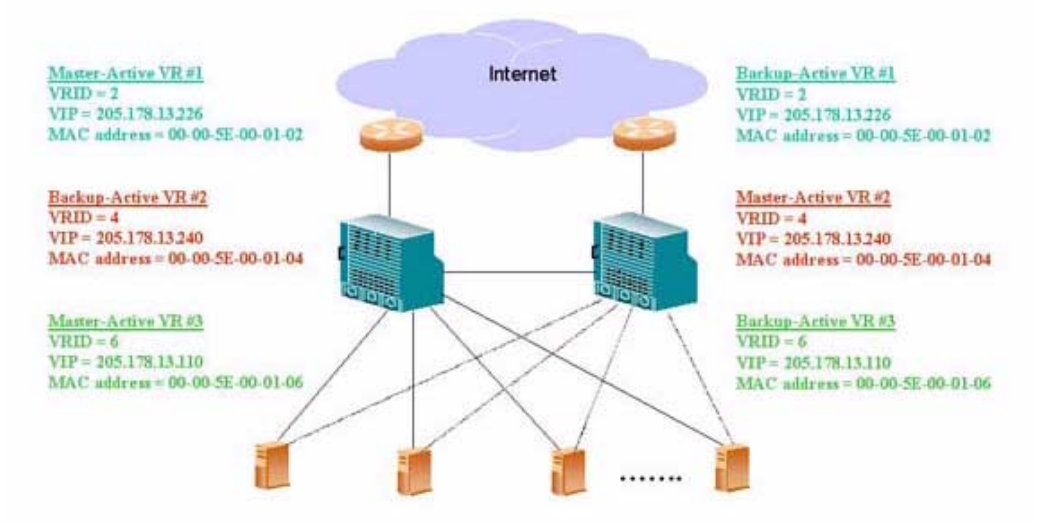


Figure 11-7 Active-Active High Availability

When sharing is used, incoming packets are processed by the switch on which they enter the virtual router. The ingress switch is determined by external factors, such as routing and Spanning Tree configuration.

NOTE – Sharing cannot be used in configurations where incoming packets have more than one entry point into the virtual router—for example, where a hub is used to connect the switches.

When sharing is enabled, the master election process still occurs. Although the process does not affect which switch processes packets that must be routed or that are destined for the virtual server IP address, it does determine which switch sends advertisements and responds to ARP requests sent to the virtual router's IP address.

Alteon WebSystems strongly recommends that sharing, rather than active-standby configurations, be used whenever possible. Sharing offers both better performance and fewer service interruptions in the face of fault conditions than active-standby configurations.

Tracking VRRP Router Priority

Web OS supports a tracking function that dynamically modifies the priority of a VRRP router, based on its current state. The objective of tracking is to have, whenever possible, the master bidding processes for various virtual routers in a LAN converge on the same switch. Tracking ensures that the selected switch is the one that offers optimal network performance. For tracking to have any effect on virtual router operation, preemption must be enabled.

NOTE – Tracking only affects hot standby and active-standby configurations. It does not have any effect on active-active sharing configurations.

Web OS can track the attributes listed in Table 11-1:

Table 11-1 VRRP Tracked Parameters

Parameter	Description
Number of virtual routers in master mode on the switch `/cfg/vrrp/track/vrs`	Useful for ensuring that traffic for any particular client/server pair is handled by the same switch, increasing routing and load-balancing efficiency. This parameter influences the VRRP router's priority in both virtual interface routers and virtual server routers.
Number of IP interfaces active on the switch `/cfg/vrrp/track/ifs`	Helps elect the virtual routers with the most available routes as the master. (An IP interface is considered active when there is at least one active port on the same VLAN.) This parameter influences the VRRP router's priority in both virtual interface routers and virtual server routers.
Number of active ports on the same VLAN `/cfg/vrrp/track/ports`	Helps elect the virtual routers with the most available ports as the master. This parameter influences the VRRP router's priority in both virtual interface routers and virtual server routers.
Number of physical switch ports that have active Layer 4 processing on the switch `/cfg/vrrp/track/l4pts`	Helps elect the main Layer 4 switch as the master. This parameter influences the VRRP router's priority in both virtual interface routers and virtual server routers.

Table 11-1 VRRP Tracked Parameters

Parameter	Description
Number of healthy real servers behind the virtual server IP address that is the same as the IP address of the virtual server router on the switch `/cfg/vrrp/track/reals`	Helps elect the switch with the largest server pool as the Master, increasing Layer 4 efficiency. This parameter influences the VRRP router's priority in virtual server routers only.
In networks where the Hot Standby Router Protocol (HSRP) is used for establishing router failover, the number of Layer 4 client-only ports that receive HSRP advertisements `/cfg/vrrp/track/hsrp`	Helps elect the switch closest to the master HSRP router as the master, optimizing routing efficiency. This parameter influences the VRRP router's priority in both virtual interface routers and virtual server routers.

Each tracked parameter has a user-configurable weight associated with it. As the count associated with each tracked item increases (or decreases), so does the VRRP router's priority, subject to the weighting associated with each tracked item. If the priority level of a backup is greater than that of the current master, then the backup can assume the role of the master.

See "Configuring the Switch for Tracking" on page 202 for an example on how to configure the switch for tracking VRRP priority.

High Availability Configurations

Alteon WebSystems switches offer flexibility in implementing redundant configurations. This section discusses a few of the more useful and easily deployed configurations:

- "Active-Standby Virtual Server Router Configuration" on page 185
- "Active-Active VIR and VSR Configuration" on page 187
- "Active/Active Server Load Balancing Configuration" on page 189
- "VRRP-Based Hot-Standby Configuration" on page 197

Active-Standby Virtual Server Router Configuration

Figure 11-8 shows an example configuration where two Alteon Web switches are used as VRRP routers in an active-standby configuration, implementing a virtual server router. Active-standby redundancy should be used in configurations that cannot support sharing, that is, configurations where incoming packets will be seen by more than one switch, such as instances where a hub is used to connect the switches. In this configuration, when both switches are healthy, only the master responds to packets sent to the virtual server IP address.

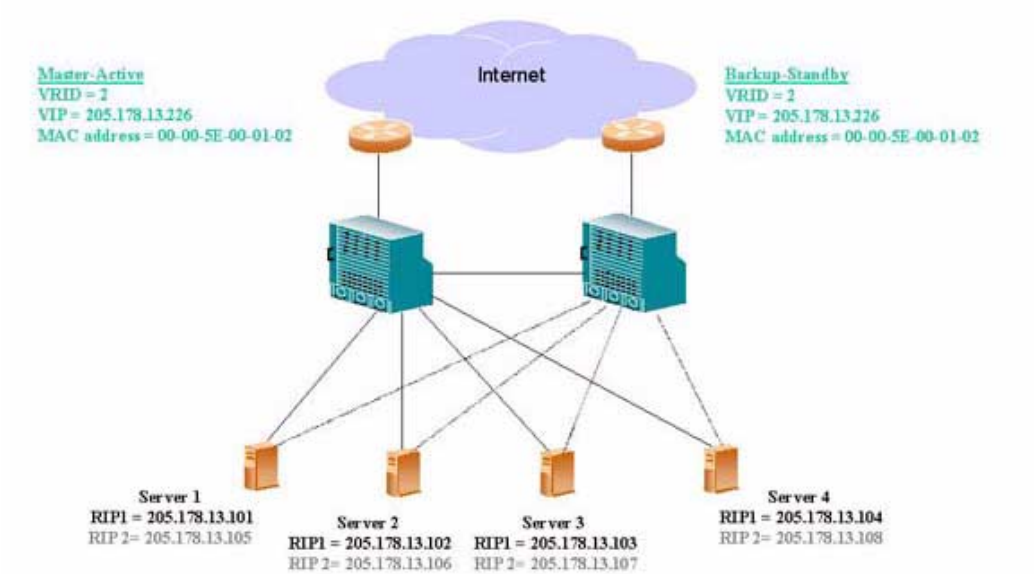


Figure 11-8 Active-Standby High-Availability Configuration

Although this example shows only two switches, there is no limit on the number of switches used in a redundant configuration. It is possible to implement an active-standby configuration across all the VRRP-capable switches in a LAN.

Each VRRP-capable switch in an active-standby configuration is autonomous. Switches in a virtual router need not be identically configured.

To implement the active-standby example, perform the following switch configuration:

1. **Configure the appropriate Layer 2 and Layer 3 parameters on both switches.**

   This includes any required VLANs, IP interfaces, default gateways, and so on. If IP interfaces are configured, none should use the virtual server IP address described in Step 4.

2. **Define all filters required for your network configuration.**

   Filters may be configured on one switch and pushed to the other switch (see Step 5, below).

3. **Configure all required SLB parameters on the left-hand switches.**

   For the purposes of this example, assume that the switch on the left in Figure 11-8 is configured in this step. Required Layer 4 parameters include a VIP = 205.178.13.226 and one real server group with four real servers, RIP = 205.178.13.101, RIP = 205.178.13.102, RIP = 205.178.13.103, and RIP = 205.178.13.104.

4. **Configure the VRRP parameters on the left-hand switch.**

   This configuration includes VRID = 2, VIP = 205.178.13.226 and the priority. Enable tracking and set the parameters appropriately (refer to "Configuring the Switch for Tracking" on page 202). Make sure to disable sharing.

5. **Synchronize the SLB and VRRP configurations by pushing the configuration from the switch on the left to the one on the right.**

   Use the `/oper/slb/synch` command.

   For more information on synchronizing switches, see "Synchronizing Configurations" on page 203.

6. **Change the real servers in the right-hand switch's configuration to RIP = 205.178.13.105, RIP = 205.178.13.106, RIP =205.178.13.107, and RIP = 205.178.13.108.**

   Adjust the right-hand switch's priority appropriately (see "Configuring the Switch for Tracking" on page 202).

   In this example, with the left-hand switch as the master, if a link between the left-hand switch and a server fails, the server will fail health checks and be taken out of the load-balancing algorithm. If that tracking is enabled and is configured to take into account the number of healthy real servers for the Virtual Router's virtual server IP address, the left-hand switch's priority will be reduced. If it is reduced to a value lower than the right-hand switch's priority, the right-hand switch will assume the role of Master.

NOTE – In this case, all active connections serviced by the left-hand switch's virtual server IP address are severed.

If the link between the left-hand (master) switch and its Internet router fails, the protocol used to distribute traffic between the routers, for example Open Shortest Path First (OSPF), will reroute traffic to the other router. The right-hand (backup) switch will act as a Layer 2/3 switch and forward all traffic destined to the virtual server IP address to the left-hand switch.

If the entire left-hand (master) switch fails, the protocol used to distribute traffic between the routers, such as OSPF, will reroute traffic to the right-hand router. The right-hand (backup) switch detects that the master has failed because it will stop receiving advertisements. The backup then assumes the master's responsibility of responding to ARP requests and issuing advertisements.

Active-Active VIR and VSR Configuration

Figure 11-9 two Alteon Web switches are used as VRRP routers in an active-active configuration implementing a virtual server router. As noted earlier, this is the preferred redundant configuration.

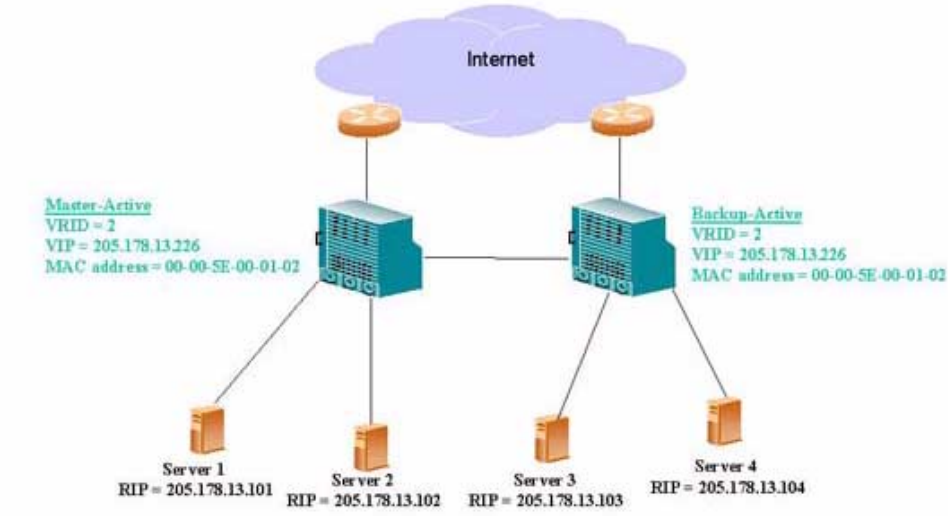


Figure 11-9 Active-Active High-Availability Configuration

Although this example shows only two switches, there is no limit on the number of switches used in a high availability configuration. It is possible to implement an active-active configuration and perform load sharing between all of the VRRP-capable switches in a LAN.

In this configuration, when both switches are healthy, the load balanced packets are sent to the virtual server IP address, resulting in higher capacity and performance than when the switches are used in an active-standby configuration.

The switch on which a frame enters the virtual server router is the one that processes that frame. The ingress switch is determined by external factors, such as routing and STP settings.

NOTE – Each VRRP-capable switch is autonomous. There is no requirement that the switches in a virtual router be identically configured. Different switch models with different numbers of ports and different enabled services may be used in a virtual router.

To implement this example, configure the switches as follows:

1. **Configure the appropriate Layer 2 and Layer 3 parameters on both switches.**

   This configuration includes any required VLANs, IP interfaces, default gateways, and so on. If IP interfaces are configured, none of them should use the virtual server IP address described in Step 4.

2. **Define all filters required for your network configuration.**

   Filters may be configured on one switch and pushed to the other switch (see Step 5, below).

3. **Configure all required SLB parameters on one of the switches.**

   For the purposes of this example, assume that the switch on the left (see Figure 11-9 on page 187) is configured in this step. Required Layer 4 parameters include a VIP = 205.178.13.226 and one real server group with two real servers, RIP = 205.178.13.101 and RIP = 205.178.13.102.

   RIP = 205.178.13.103 should be configured as a backup server to RIP = 205.178.13.101.

   RIP = 205.178.13.104 should be configured as a backup server to RIP = 205.178.13.102.

NOTE – In this configuration, each server's backup is attached to the other switch. This ensures that operation will continue if all of the servers attached to a switch fail.

---

4. **Configure the VRRP parameters on the switch.**

   Configure VRID = 2, VIP address = 205.178.13.226, and priority. Be sure to enable sharing.

5. **Synchronize the SLB and VRRP configurations by pushing the configuration from the switch on the left to the one on the right.**

   Use the `/oper/slb/sync` command.

6. **Reverse the roles of the real servers and their backups in the right switch's configuration.**

   Configure RIP = 205.178.13.103 and RIP= 205.178.13.104 as the real servers
   Configure RIP = 205.178.13.101 and RIP = 205.178.13.102 as their backups, respectively.

   In this configuration, if a link between a switch and a server fails, the server will fail health checks and its backup (attached to the other switch) will be brought online. If a link between a switch and its Internet router fails, the protocol used to distribute traffic between the routers (for example, OSPF) will reroute traffic to the other router. Since all traffic now enters the virtual server router on one switch, that switch will process all incoming connections.

   If an entire master switch fails, the backup will detect this failure because it will stop receiving advertisements. The backup will assume the master's responsibility of responding to ARP requests and issuing advertisements.

Be cautious before setting the maximum connections (`maxconn`) metric in this configuration. The `maxcon` number is not shared between switches. Therefore, if a server is used for normal operation by one switch and is activated simultaneously as a backup by the other switch, the total number of possible connections to that server will be the sum of the maximum connection limits defined for it on both switches.

Active/Active Server Load Balancing Configuration

In this example, you set up four virtual servers each load balancing two servers providing one service (for example, HTTP) per virtual server.

You are load balancing HTTP, HTTPS, POP3, SMTP, and FTP. Each protocol is load balanced via a different virtual server. You could load balance all of these services on one virtual server IPaddress, but in this example, four distinct virtual servers are used to illustrate the benefits of active/active failover. Set up one switch, dump out the configuration script (also called a text dump), edit it, and dump the edited configuration into the peer switch.

NOTE – Configuring the switch for active-active failover should take at least 15 minutes to complete. You can use either the Web OS Browser-Based Interface (BBI) or the Command Line Interface (CLI) for configuration.

Procedure #1: Background Configuration

1. **Define the IP interfaces.**

The switch will need an IP interface for each subnet to which it will be connected so it can communicate with devices attached to it. Each interface will need to be placed in the appropriate VLAN. In our example, Interfaces 1, 2, 3, and 4 will be in VLAN 2 and Interface 5 will be in VLAN 1.

NOTE – On Alteon Web switches, you may configure more than one subnet per VLAN.

To configure the IP interfaces for this example, enter the following commands from the CLI:

```
>> Main# /cfg/ip/if 1                    (Select IP interface 1)
>> IP Interface 1 # addr 10.10.10.10     (Assign IP address for the interface)
>> IP Interface 1 # vlan 2               (Assign VLAN for the interface)
>> IP Interface 1 # ena                  (Enable IP interface 1)
```

Repeat the commands for each interface listed below:

- IF 2—20.10.10.10
- IF 3—30.10.10.10
- IF 4—40.10.10.10
- IF 5—200.1.1.10

2. Define the VLANs.

In this configuration, set up two VLANs: One for the outside world (the ports connected to the upstream switches, toward the routers) and one for the inside (the ports connected to the downstream switches, toward the servers).

```
>> Main# /cfg/vlan <VLAN number>          (Select VLAN 1)
>> vlan 1 # add <port number>             (Add a port to the VLAN membership)
>> vlan 1 # ena                           (Enable VLAN 1)
```

Repeat this command for the second VLAN.

- VLAN 1 - IF 5—physical ports connected to upstream switches.
- VLAN 2 - IFs 1,2,3,4—physical ports connected to downstream switches.

3. Disable Spanning Tree.

If you have configured multiple spanning trees, specify the spanning tree group number.

```
>> Main# /cfg/stp 1                       (Select the STP Group number)
>> Main# /cfg/stp/off                     (Disable STP)
>> Main# /cfg/stp/apply                   (Make your changes active)
```

4. Enable IP forwarding.

IP forwarding is enabled by default. Make sure IP forwarding is enabled if the virtual IP addresses and real IP addresses are on different subnets, or if the switch is connected to different subnets and those subnets need to communicate through the switch. If you are in doubt as to whether or not to enable IP forwarding, enable it. In this example, the virtual IP addresses and real IP addresses are on different subnets, so enable this feature using the following command:

```
>> Main# /cfg/ip/frwd/on                  (Enable IP forwarding)
```

Procedure #2: SLB Configuration

1. **Define the Real Servers.**

   The real server IP addresses are defined and put into four groups, depending on the service they are running. Notice that RIPs 7 and 8 are on routable subnets in order to support passive FTP. For each real server, you must assign a real server number, specify its actual IP address, and enable the real server. For example:

```
>> Main# /cfg/slb/real 1                    (Server A is real server 1)
>> Real server 1 # rip 10.10.10.5/24        (Assign Server A IP address)
>> Real server 1 # ena                      (Enable real server 1)
```

   Repeat this sequence of commands for the following real servers:

   - RIP 2—10.10.10.6/24
   - RIP 3—20.10.10.5/24
   - RIP 4—20.10.10.6/24
   - RIP 5—30.10.10.5/24
   - RIP 6—30.10.10.6/24
   - RIP 7—200.1.1.5/24
   - RIP 8—200.1.1.6/24

2. **Define the real server groups, adding the appropriate real servers.**

   This combines the three real servers into one service group:

```
>> Real server 8 # /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)
```

   Repeat this sequence of commands for the following real server groups:

   - Group 2—Add RIP 3 and 4
   - Group 3—Add RIP 5 and 6
   - Group 4—Add RIP 7 and 8

3. **Define the virtual servers.**

After defining the virtual server IP addresses and associating them with a real server group number, you must tell the switch which IP ports/services/sockets you want to load balance on each virtual server IP address. You can specify the service by either the port number, service name, or socket number.

```
>> Real server group 4 # /cfg/slb/virt 1          (Select virtual server 1)
>> Virtual server 1 # vip 200.200.200.100        (Assign a virtual server IP address)
>> Virtual Server 1 # service 80                 (Assign HTTP service port 80)
>> Virtual server 1 http Service # group 1       (Associate virtual port to real group)
>> Virtual server 1 # ena                        (Enable the virtual server)
```

Repeat this sequence of commands for the following virtual servers:

- VIP 2 200.200.200.101 will load balance HTTPS (Port 443) to Group 2
- VIP 3 200.200.200.102 will load balance POP/SMTP (Ports 110/25) to Group 3
- VIP 4 200.200.200.104 will load balance FTP (Ports 20/21) to Group 4

4. **Define the client and server port states.**

Defining a client port state tells that port to watch for any frames destined for the virtual server IP address and to load balance them if they are destined for a load-balanced service. Defining a server port state tells the port to the do the remapping (NAT) of the real server IP address back to the virtual server IP address. Note the following:

- The ports connected to the upstream switches (the ones connected to the routers) will need to be in the client port state.
- The ports connected to the downstream switches (the ones providing fan out for the servers) will need to be in the server port state.

Configure the ports, using the following sequence of commands:

```
>> Virtual server 4# /cfg/slb/port a1          (Select physical switch port a1)
>> SLB port A1 # client ena                    (Enable client processing on port a1)
>> SLB port A1 # ../port a2                    (Select physical switch port a2)
>> SLB port A2 # server ena                    (Enable server processing on port a2)
```

Procedure #3: Virtual Router Redundancy Configuration

1. **Configure virtual routers 2, 4, 6, and 8.**

   These virtual routers will have the same IP addresses as the virtual server IP address. This is what tells the switch that these are virtual service routers (VSRs). In this example, Layer 3 bindings are left in their default configuration, which is disabled.

   Configure a virtual router using the following sequence of commands:

```
>> Virtual server 4 # /cfg/vrrp/vr 2          (Select virtual router 2)
>> Virtual router 2 vrid 2                    (Set virtual router ID)
>> Virtual router 2 addr 200.200.200.100      (Assign virtual router IP address)
>> Virtual router 2 if 5                      (Assign virtual router interface)
>> Virtual router 2 ena                       (Enable virtual router 2)
```

   Repeat this sequence of commands for the following virtual routers:

   - VR 4 - VRID 4 - IF 5 (associate with IP interface 5)—Address 200.200.200.101
   - VR 6 - VRID 6 - IF 5 (associate with IP interface 5)—Address 200.200.200.103
   - VR 8 - VRID 8 - IF 5 (associate with IP interface 5)—Address 200.200.200.104

2. **Configure virtual routers 1, 3, 5, and 7.**

   These virtual routers will act as the default gateways for the servers on each respective subnet. Because these virtual routers are survivable next hop/default gateways, they are called virtual interface routers (VIRs).

   Configure each virtual router listed below, using the sequence of commands in Step 1.

   - VR 1—VRID 1 - IF 1 (associate with IP interface 1) – Address 10.10.10.1
   - VR 3—VRID 3 - IF 2 (associate with IP interface 2) – Address 20.10.10.1
   - VR 5—VRID 5 - IF 3 (associate with IP interface 3) – Address 30.10.10.1
   - VR 7—VRID 7 - IF 4 (associate with IP interface 4) – Address 40.10.10.1

3. **Set the renter priority for each virtual router.**

   Since you want Switch 1 to be the master router, you need to bump the default virtual router priorities (which are 100 to 101 on virtual routers 1-4) to force switch 1 to be the master for these virtual routers.

   Use the following sequence of commands:

```
>> Virtual server 4 # /cfg/vrrp/vr 1          (Select virtual router 1)
>> Virtual router 1 prio 101                  (Set virtual router priority)
```

   Apply this sequence of commands to the following virtual routers, assigning each a priority of 101:

   - VR 2 - Priority 101
   - VR 3 - Priority 101
   - VR 4 - Priority 101

4. **Configure priority tracking parameters for each virtual router.**

   For this example, the best parameter(s) on which to track is Layer 4 ports (l4pts).

   Use the following command:

```
>> Virtual server 4# /cfg/vrrp/vr 1/track l4pts
```

   This command sets the priority tracking parameter for virtual router 1, electing the virtual router with the most available ports as the master router. Repeat this command for the following virtual routers:

   - VR 2—Track l4pts
   - VR 3—Track l4pts
   - VR 4—Track l4pts
   - VR 6—Track l4pts
   - VR 7—Track l4pts
   - VR 8—Track l4pts

   Switch 1 configuration is complete.

Configuring Switch 2

Use the following procedure to dump the configuration script (text dump) out of Switch 1:

- Using the Browser Based Interface (BBI)

  (a) You need a serial cable that is a DB-9 Male to DB-9 Female, straight-through (not a null modem) cable.

  (b) Connect the cable from a COM port on your computer to the console port on switch 1.

  (c) Open HyperTerminal (or the terminal program of your choice) and connect to the switch using the following parameters: Baud: 9600, Data Bits: 8, Parity: None, Stop Bits:1, Flow Control: None.

- Using HyperTerminal

  (a) Only the Baud Rate and Flow Control options need to be changed from the default settings.

  (b) Once you connect to the switch, start logging your session in HyperTerminal (transfer/capture text).

  (c) Save the file as "Customer Name" Switch 1, then type the following command in the switch command line interface:

  `/cfg/dump`

  A script will be dumped out.

  (d) Stop logging your session (transfer/capture text/stop).

Modify the script as follows:

1. **Open the text file that you just created and change the following:**
   - Delete anything above "Script Start."
   - Delete the two lines directly below "Script Start." These two lines identify the switch from which the dump was taken and the date and time. If these two lines are left in, it will confuse Switch 2 when you dump in the file.
   - Change the last octet in all the IP interfaces from .10 to .11. Find this in line: `/cfg/ip/if 1/addr 10.10.10.10`. Simply delete the "0" and put in a "1." Be sure to do this for all the IP interfaces, otherwise you will have duplicate IP addresses in the network.

2. **Change the virtual router priorities.**

   Virtual routers 1–4 need to have their priority set to 100 from 101, and virtual routers 5-7 need to have their priorities set to 101 from 100. You can find this in line `/cfg/vrrp/vr 1/vrid 1/if 1/prio 101`.

3. **Scroll to the bottom of the text file and delete anything past “Script End.”**

4. **Save the changes to the text file as “Customer Name” Switch 2.**

   Move your serial cable to the console port on the second switch. Any configuration on it needs to be deleted by resetting it to factory settings, using the following command:

   ```
   >> Main# /boot/conf factory/reset
   ```

   You can tell if the switch is at factory default when you log on because the switch will prompt you if you want to use the step-by-step configuration process. When it does, respond: “No.”

5. **In HyperTerminal, go to transfer/send text file and send the switch 2 text file. The configuration will dump into the switch. Simply type “apply,” then “save.” When you can type characters in the terminal session again, reboot the switch (`/boot/reset`).**

VRRP-Based Hot-Standby Configuration

A hot-standby configuration allows all processes to failover to a backup switch if any type of failure should occur. The primary application for hot-standby redundancy is to avoid bridging loops when using the Spanning Tree Protocol (STP), IEEE 802.1d.

NOTE – To use hot-standby redundancy, peer switches must have an equal number of ports.

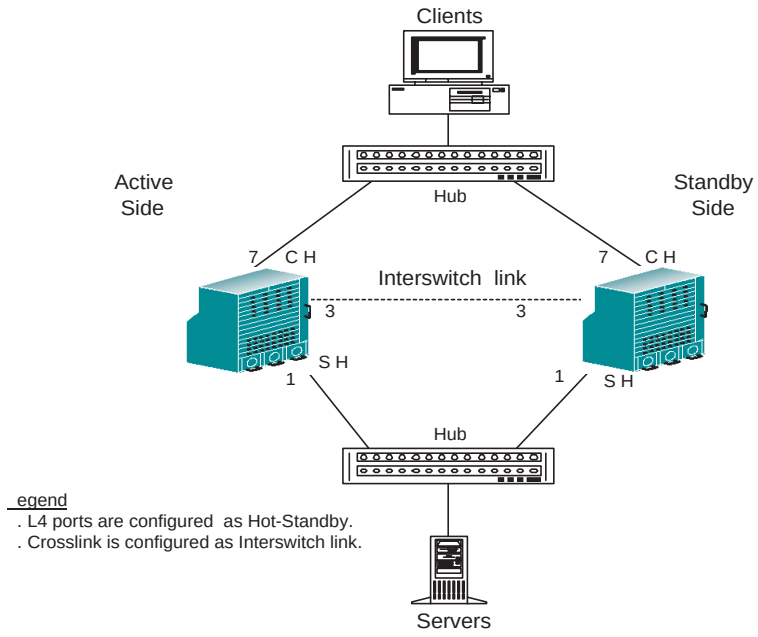


Figure 11-10 Hot-Standby Configuration

Figure 11-10 shows a classic network topology, designed with redundancy in mind. This topology contains bridging loops that would require the use of STP. In the typical network, STP failover time is 45-50 seconds, much longer than the typical failover rate using VRRP only.

NOTE – In complex networks, STP convergence time can be much higher than 45-50 seconds.

If VRRP was used in this configuration, it would require STP. An important factor to consider is that the switch would be affected by the slower failover time of STP even if VRRP were in use.

While VRRP can be used without STP in this scenario, doing so would involve a more complex network configuration, requiring multiple subnets and/or VLANs and enabling IP forwarding to route between them.

By reducing complexity to a single subnet and not requiring routing (Layer 3), hot-standby can be used. The key to hot-standby is that the *interswitch link* (the link between switches), does *not* participate in STP, so there are no loops in the topology (see Figure 11-10). STP does not need to be enabled, and the switch will have failover times similar to what would be the case with VRRP.

Configuration Procedure

Configuration takes place after configuring SLB and VRRP with STP enabled:

1. **From the SLB menu, enable a hot-standby link on the Layer 4 ports; then enable inter-switch link on the crosslink.**

```
>> Main# /cfg/slb/port a1                    (Select port a1)
>> SLB Port A1# server ena                   (Enable the server)
>> SLB Port A1# hotstan ena                  (Enable hotstandby)
>> SLB Port A1# ../port a2                   (Select port a2)
>> SLB Port A2# hotstan ena                  (Enable hotstandby)
>> SLB Port A2# ../port a3                   (Select port a3)
>> SLB Port A3# intersw ena                  (Enable inter-switch processing)
>> SLB Port A3# ../port a7                   (Select port a7)
>> SLB Port A7# client ena                   (Enable the client)
>> SLB Port A7# hotstan ena                  (Enable hotstandby)
```

2. **From the VRRP menu, enable VRRP group mode; then enable hot-standby.**

```
>> Main# /cfg/vrrp/on                                    (Enable VRRP)
>> Virtual Router Redundancy Protocol# hotstan ena       (Enable hotstandby)
>> Virtual Router Redundancy Protocol# group ena         (Enable VR group)
```

3. **Sync the VRRP, SLB and filter settings to the other switch (same ports)**.

```
>> Main# /oper/slb/sync
```

NOTE – Switches peering with each other must have an equal number of ports.

4. **Turn off STP after verifying that the network is stable.**

```
>> Main# /cfg/stp 1/off                    (Disable STP group)
>> Spanning Tree Group 1# save             (Enable hotstandby)
>> Main# /boot/reset                       (Reset the switch)
```

NOTE – You must reboot the switch for the hot-standby configuration to take effect.

Virtual Router Deployment Considerations

Review the following issues described in this section to prevent network problems when deploying virtual routers:

- Mixing Active-Standby and Active-Active Virtual Routers
- Synchronizing Active/Active Failover
- Eliminating Loops with STP and VLANs
- Assigning VRRP Virtual Router ID
- Configuring the Switch for Tracking
- Synchronizing Configurations

Mixing Active-Standby and Active-Active Virtual Routers

If the network environment can support sharing, enable it for all virtual routers in the LAN. If not, use active-standby for all virtual routers. Do not mix active-active and active-standby virtual routers in a LAN. Mixed configurations have not been tested, may result in unexpected operational characteristics, and, therefore, are not recommended.

Synchronizing Active/Active Failover

The hot-standby failover required the primary and secondary switches to have identical configurations and port topology. With VRRP and active/active failover, this is optional. Each switch can be configured individually with different port topology, SLB, and filters. If you would rather force two active/active switches to use identical settings, you can synchronize their configuration using the following command:

```
/oper/slb/synch
```

The sync command copies the following settings to the switch at the specified IP interface address:

- VRRP settings
- SLB settings (including port settings)
- Filter settings (including filter port settings)
- Proxy IP settings

If you perform the sync command, you should check the configuration on the target switch to ensure that the settings are correct.

For more information on synchronizing configurations between two switches, see "Synchronizing Configurations" on page 203.

Eliminating Loops with STP and VLANs

VRRP active/active failover is significantly different from the hot-standby failover method supported in previous releases. As shown in Figure 11-11, active-active configurations can introduce loops into complex LAN topologies.

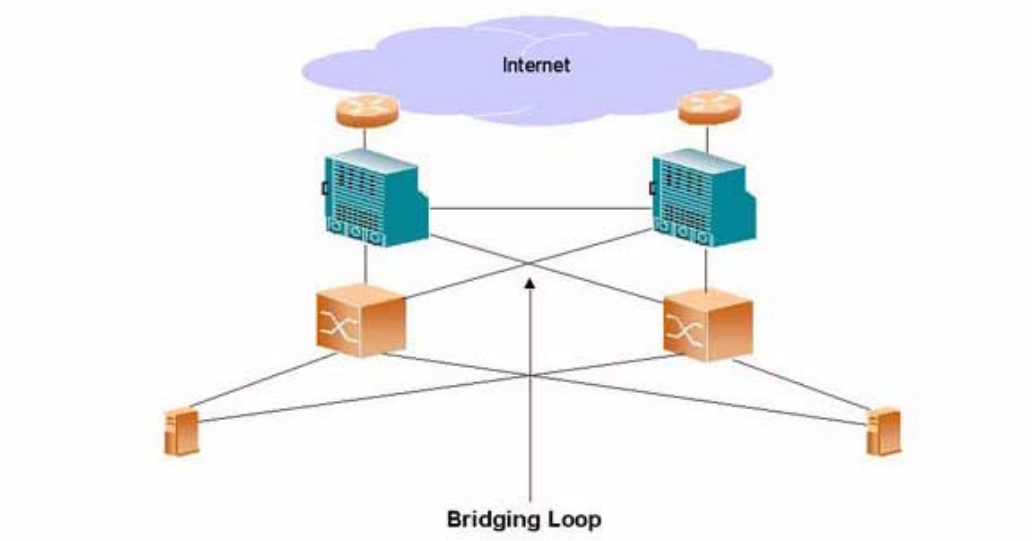


Figure 11-11 Loops in Active-Active Configuration

Using STP to Eliminate Loops

VRRP generally requires Spanning Tree Protocol to be enabled in order to resolve bridge loops that usually occur in cross-redundant topologies, as shown in Figure 11-12. In this example, a number of loops are wired into the topology. STP resolves loops by blocking ports where looping is detected.

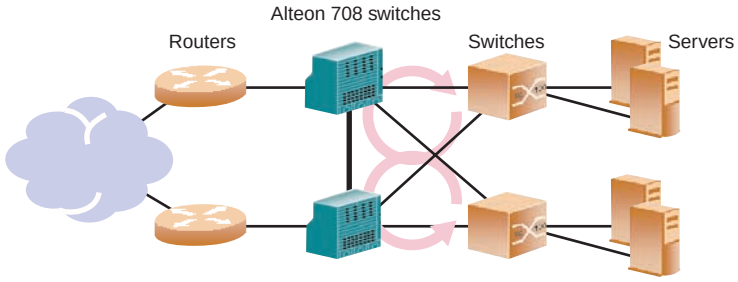


Figure 11-12 Cross-Redundancy Creates Loops, but STP Resolves Them

One drawback to using STP with VRRP is the failover response time. STP could take as long as 45 seconds to re-establish alternate routes after a switch or link failure.

Using VLANs to Eliminate Loops

When using VRRP, you can decrease failover response time by using VLANs instead of STP to separate traffic into non-looping broadcast domains. An example is shown in Figure 11-13:

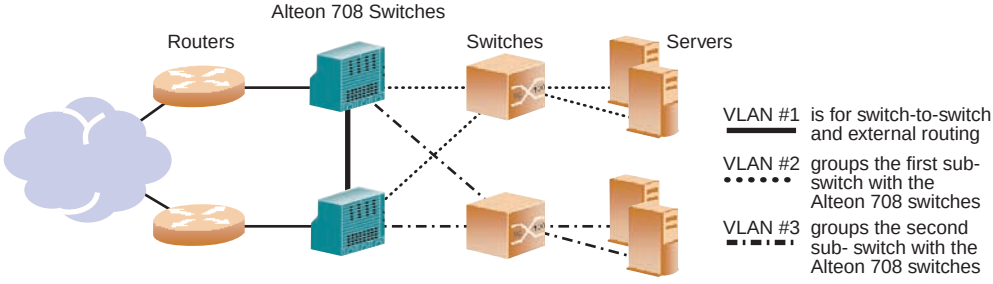


Figure 11-13 Using VLANs to Create Nonlooping Topologies

This topology allows STP to be disabled. On the Alteon 708 switches, IP routing allows traffic to cross VLAN boundaries. The servers use the Alteon 708 switches as default gateways. For port failure, traffic is rerouted to the alternate path within one health-check interval (configurable between 1 and 60 seconds, with a default of 2 seconds).

Assigning VRRP Virtual Router ID

During the software upgrade process, VRRP virtual router IDs will be automatically assigned if failover is enabled on the switch. When configuring virtual routers at any point after upgrade, virtual router ID numbers (`/cfg/vrrp/vr #/vrid`) must be assigned. The virtual router ID may be configured as any number between 1 and 255.

Configuring the Switch for Tracking

Tracking configuration largely depends on user preferences and network environment. Consider the configuration shown in Figure 11-8 on page 185. Assume that the user wants the following behavior on the network:

- The switch on the left will be the master router upon initialization.
- If the switch on the left is the master and it has one active server fewer than the switch on the right, it remains the master.
- The user wants this behavior because the user believes running one server down is less disruptive than bringing a new master online and severing all active connections in the process.
- If the switch on the left is the master and it has two or more active servers fewer than the switch on the right, the switch on the right becomes the master.
- If the switch on the right is the master, it remains the master even if servers are restored to the point on the left-hand switch where the left-hand switch has one fewer or an equal number of servers.
- If the switch on the right is the master and it has one active server fewer than the switch on the left, the switch on the left becomes the master.

The user can implement this behavior by configuring the switch for tracking as follows:

1. **Set the priority for the left-hand switch to the default value of 100.**
2. **Set the priority for the right-hand switch to 96.**
3. **On both switches, enable tracking based on the number of virtual routers in master mode on the switch and set the value = 5.**
4. **On both switches, enable tracking based on the number of healthy real servers behind the virtual server IP address that is the same as the IP address of the virtual server router on the switch and set the value = 6.**

   Initially, the switch (Figure 11-8) on the left will have a priority of 100 (base value) + 5 (since it will initially be the master) + 24 (4 active real servers x 6 per real server) = 129. The switch on the right with have a priority of 96 (base value) + 24 (4 active real servers X 6 per real server) = 120.

If one server attached to the left-hand switch fails, the left-hand switch's priority will be reduced by 6 to 123. Since 123 is greater than 120 (the right-hand switch's priority), the left-hand switch will remain the master.

If a second server attached to the left-hand switch fails, the left-hand switch's priority will be reduced by 6 more to 117. Since 117 is less than 120 (the right-hand switch's priority), the right-hand switch will become the Master. At this point, the left-hand switch's priority will fall by 5 more and the right-hand switch's will rise by 5 because the switches are tracking how many Masters they are running. So, the left-hand switch's priority will settle out at 112 and the right-hand switch's priority at 125.

When both servers are restored to the left-hand switch, that switch's priority will rise by 12 (2 healthy real servers X 6 per healthy server) to 124. Since 124 is less than 125, the right-hand switch will remain the master.

If, at this point, a server fails on the right-hand switch, its priority will fall by 6 to 119. Since 119 is less than 124, the left-hand switch will become the Master. Its priority will settle out at 129 (since it's now the master) while the right-hand switch's will drop by 5 more to 114.

You can see from this example that the user's goals were met by the configured tracking parameters.

NOTE – There is no shortcut to setting tracking parameters. The goals must first be set and the outcomes of various configurations and scenarios analyzed to find settings that meet the goals.

Synchronizing Configurations

As noted above, each VRRP-capable switch is autonomous. Switches in a virtual router need not be identically configured. As a result, configurations cannot be synchronized automatically.

For user convenience, it is possible to push a configuration from one VRRP-capable switch to another using the `/oper/slb/sync` command. However, care must be taken when using this command to avoid unexpected results. All server load balancing, port configurations, filter configurations, and VRRP parameters can be pushed using the `/oper/slb/synch` command.

NOTE – Before you synchronize the configuration between two switches, a peer must be configured on each switch. Switches being synchronized must use the same administrator password.

Configure the two switches as peers to each other. From Switch 1, configure Switch 2 as a peer and specify its IP address as follows:

```
>> Main # /cfg/slb/sync                    (Select the synchronization menu)
>> Config Synchronization # peer 1         (Select a peer)
>> Peer Switch 1 # addr <IP address>       (Assign Switch 2 IP address)
>> Peer Switch 1 # enable                  (Enable peer switch)
```

Similarly, from Switch 2, configure Switch 1 as a peer and specify its IP address as follows:

```
>> Main # /cfg/slb/sync                    (Select the synchronization menu)
>> Config Synchronization # peer 2         (Select a peer)
>> Peer Switch 2 # addr <IP address>       (Assign switch 1 IP address)
>> Peer Switch 2 # enable                  (Enable peer switch)
```

Port specific parameters, such as what filters are applied and enabled on what ports, are part of what is pushed by the `/oper/slb/synch` command. As a result, if the `/oper/slb/synch` command is used, it is highly recommended that the hardware configurations and network connections of all switches in the virtual router be identical. That is, each switch should be the same model, have the same line cards in the same slots (if modular) and have the same ports connected to the same external network devices. Otherwise, unexpected results may occur if the `/oper/slb/synch` command attempts to configure a non-existent port or applies an inappropriate configuration to a port.

Part 3: Advanced Web Switching

Advanced Web switching requires the ability to parse requests and classify flows using host tags so that each request can be isolated and treated accordingly. This part discusses two applications using Web OS 7.2 features.

- Global Server Load Balancing
- Firewall Load Balancing

050156A, June 2001

Alteon WebSystems
Web Speed for e-Business

CHAPTER 12

Global Server Load Balancing

This chapter provides a conceptual overview of Global Server Load Balancing (GSLB) and configuration examples for performing GSLB across multiple geographic sites. The following topics are addressed in this chapter:

- "GSLB Overview" on page 208. This section explains the benefits of GSLB and how GSLB works.
- "Configuring GSLB" on page 211. This section provides a step-by-step procedure to configure GSLB on the Alteon 708.
- "Using Proxy IP for Non-HTTP Redirects" on page 222. This section explains how client proxy IP addressing is used to redirect non-HTTP applications.
- "Verifying GSLB Operation" on page 225. This section lists commands to verify GSLB operation.
- "Configuring Client Site Preferences" on page 226. This section provides information on configuring client proximity tables to customize site preferences for client networks.
- "Using Border Gateway Protocol for GSLB" on page 229. BGP-based GSLB allows you to localize content delivery to the most efficient and consistent site.

GSLB Overview

GSLB allows you to balance server traffic load across multiple physical sites. Alteon WebSystems' GSLB implementation takes into account an individual site's health, response time, and geographic location to smoothly integrate the resources of the dispersed server sites for a complete global performance.

NOTE – The number of supported GSLB sites per switch has been increased to 128, with a total aggregate of 8192 service/site combinations.

Benefits

GSLB meets the following demands for distributed network services:

- High content availability through distributed content and distributed decision making. If one site becomes disabled, the others become aware of it and take up the load.
- There is no latency during client connection set up. Instant site hand-off decisions can be made by any distributed switch.
- The best performing sites get the majority of traffic over a given period of time but are not overwhelmed.
- Switches at different sites regularly exchange information through the Distributed Site State Protocol (DSSP) and can trigger exchanges when any site's health status changes. This ensures that each active site has valid state knowledge and statistics.
- GSLB implementation takes geography as well as network topology into account.
- Creative control is given to the network administrator or Webmaster to build and control content by user, location, target application, and more.
- GSLB is easy to deploy, manage, and scale. Switch configuration is straightforward. There are no complex system topologies involving routers, protocols, and so forth.
- Flexible design options are provided.
- All Internet Protocols are supported.

How GSLB Works

Global server load balancing is based on the Domain Name System (DNS) and proximity by source IP address. In Figure 12-1, a client is using a Web browser to view the Web site for the Foo Corporation at "www.foocorp.com." The Foo Corporation has two sites: one in California, and one in Denver, each with identical content and available services. Both sites have an Alteon 708 configured for GSLB. These switches are also configured as the Authoritative Name Servers for "www.foocorp.com."

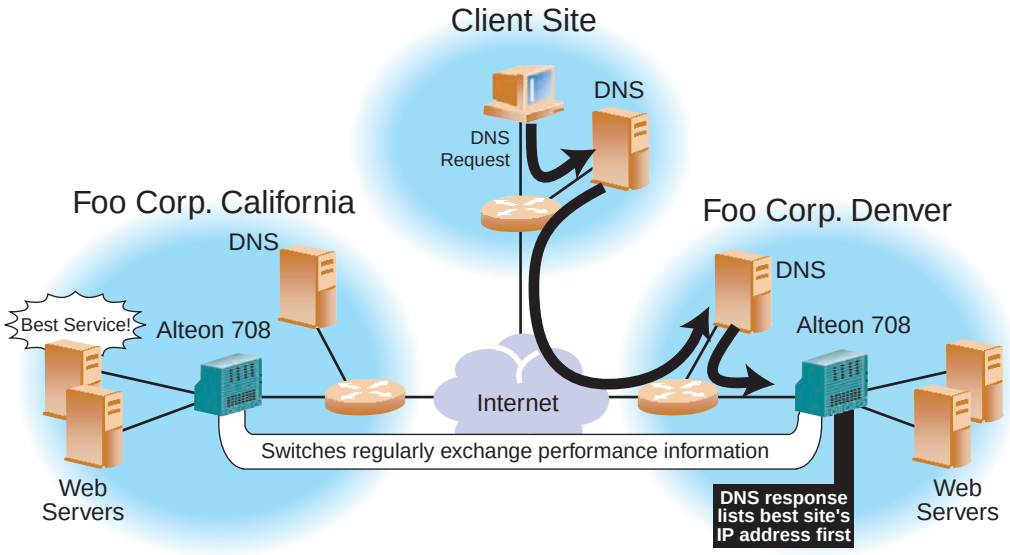


Figure 12-1 DNS Resolution with Global Server Load Balancing

The DNS resolution for GSLB is described in detail in the following procedure:

1. **The client Web browser requests the "www.foocorp.com" IP address from the local DNS.**
2. **The client's DNS asks its upstream DNS, which in turn asks the next, and so forth, until the address is resolved.**

   Eventually, the request reaches an upstream DNS server that has this information available or the request reaches one of the Foo Corporation's DNS servers.

3. **The Foo Corporation's Denver DNS has been configured to know that the local Alteon 708 with GSLB software is the authoritative name server for "www.foocorp.com."**

   However, the GSLB software knows that Foo Corp. California currently provides better service and responds with Foo Corp. California's virtual IP address.

   Each switch with GSLB software is capable of responding to the client's name resolution request. Since each switch regularly checks and communicates health and performance information with its peers, either switch can determine which site(s) are best able to serve the client's Web access needs. It can respond with a list of IP addresses for the Foo Corporation's distributed sites, prioritized by performance, geography, and other criteria.

4. **The client connects to Foo Corp. California for the best service.**

   The client's Web browser will use the IP address information to open a connection to the best available site. The IP addresses represent virtual servers at any site, which are locally load balanced according to regular SLB configuration.

   If the site serving the client HTTP content suddenly experiences a failure (no healthy real servers) or becomes overloaded with traffic (all real servers reach their maximum connection limit), the switch issues an HTTP Redirect and transparently causes the client to connect to another peer site.

   The end result is that the client gets quick, reliable service, with no latency and no special client-side configuration.

Configuring GSLB

Configuring GSLB is simply an extension of the configuration procedure for SLB. The process is summarized as follows:

- Use the administrator login to connect to the switch you want to configure.
- Activate SLB and GSLB software keys. See the *Web OS 7.2 Command Reference* for details.
- Configure the switch at each site with basic attributes.
  - Configure the switch IP interface.
  - Configure the default gateways.
- Configure the switch at each site to act as the DNS server for each service hosted on its virtual servers. Also, configure the local DNS server to recognize the switch as the authoritative DNS server for the hosted services.
- Configure the switch at each site for local SLB.
  - Define each local real server.
  - Group local real servers into real server groups.
  - Define the local virtual server with its IP address, services, and real server groups.
  - Define the switch port states.
  - Enable SLB.
- Finally, make each switch recognize its remote peers.
  - On each switch, configure a remote real server entry for each remote service.
  - Add the remote real server entry to an appropriate real server group.
  - Enable GSLB.

Consider the following example network:

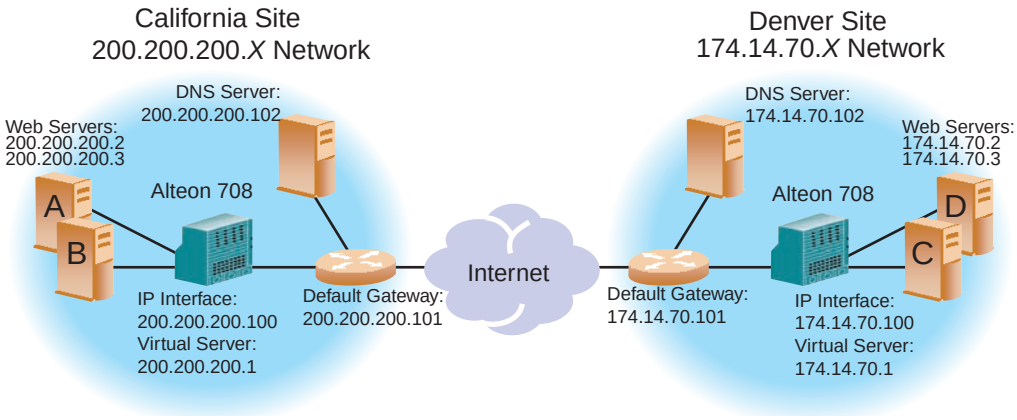


Figure 12-2 GSLB Topology Example

In the following examples, many of the options are left to their default values. See "Additional SLB Options" on page 117 for more information on configuring additional SLB options. For details about any of the processes or menu commands described in this example, see the *Web OS 7.2 Command Reference*.

The following sections provide a step-by-step procedure for configuring GSLB:

1. **"Step 1: Configure the Basics at the California Site" on page 213.**
2. **"Step 2: Configure the California Switch for Standard SLB" on page 214.**
3. **"Step 3: Configure the California Site for GSLB" on page 216.**
4. **"Step 4: Configure the Basics at the Denver Site" on page 217.**
5. **"Step 5: Configure the Denver Switch for Standard SLB" on page 218.**
6. **"Step 6: Configure the Denver Site for GSLB" on page 220.**

Step 1: Configure the Basics at the California Site

1. **If the Browser-Based Interface (BBI) is to be used for managing the California switch, change its service port.**

   GSLB uses service port 80 on the IP interface for DSSP updates. By default, the Web OS Browser-Based Interface (BBI) also uses port 80. Both services cannot use the same port. If the Web-based interface is enabled (see the `/cfg/sys/web` command in Chapter 6 of the *Web OS Command Reference*), configure it to use a different port.

   Enter the following command to change the BBI port to use 8080:

```
>> # /cfg/sys                          (Select the System Menu)
>> System# wport 8080                  (Set service port 8080 for BBI)
```

2. **On the California switch, define an IP interface.**

   The IP interface resolves client DNS requests. The IP interface must have an IP route to the local real servers. The switch uses this route to determine if the real servers can be reached via TCP/IP.

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

```
>> System# /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# ena                       (Enable IP interface 1)
```

NOTE – This example assumes that all ports and IP interfaces use default VLAN 1, requiring no special VLAN configuration for the ports or IP interface.

3. **On the California switch, define the default gateway.**

   The router at the edge of the site acts as the default gateway to the Internet. To configure the default gateway for this example, enter these commands from the CLI:

```
>> IP Interface 1# ../gw 1                   (Select default gateway 1)
>> Default gateway 1# addr 200.200.200.101  (Assign IP address for the gateway)
>> Default gateway 1# ena                    (Enable default gateway 1)
```

4. **Configure the local DNS server to recognize the local GSLB switch as the authoritative name server for the hosted services.**

   Determine the domain name that will be distributed to both sites and the host name for each distributed service. In this example, the California DNS server is configured to recognize 200.200.200.100 (the IP interface of the California GSLB switch) as the authoritative name server for “www.foocorp.com.”

Step 2: Configure the California Switch for Standard SLB

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

   The real servers in any 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 Requirements" on page 110.

   For this example, the host real servers have IP addresses on the same IP subnet:

   Table 12-1 GSLB Example: California Real Server IP Addresses

Real Server	IP address
Server A	200.200.200.2
Server B	200.200.200.3

2. **On the California switch, define each local real server.**

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

```
>> Default gateway 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 # ena                         (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 # ena                         (Enable real server 2)
```

3. **On the California switch, define a real server group.**

   Combine the real servers into one service group and set the necessary health checking parameters.

   In this example, HTTP health checking is used to ensure that Web content is being served. If the `index.html` file is not accessible on a real server during health checks, the real server will be marked as down.

```
>> Real server 2 # ../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# health http            (Use HTTP for health checks)
>> Real server group 1# content index.html     (Set URL content for health checks)
```

4. **On the California switch, define a virtual server.**

   All client requests will be addressed to a virtual server IP address defined on the switch. Clients acquire the virtual server IP address through normal DNS resolution. HTTP uses well-known TCP port 80. In this example, HTTP is configured as the only service running on this virtual server IP address and is associated with the real server group. For example:

```
>> Real server group 1 # ../virt 1              (Select virtual server 1)
>> Virtual server 1# vip 200.200.200.1         (Assign a virtual server IP address)
>> Virtual Server 1# service 80
>> Virtual server 1 http Service# group 1      (Associate virtual port to real group)
>> Virtual server 1 http Service# ../ena       (Enable virtual server)
```

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 Table 8-2 on page 131.

5. **On the California switch, set the type of Layer 4 traffic processing each port must support.**

   In this example, the following ports are used on the Alteon 708 Web switch:

Table 12-2 GSLB Example: California Alteon 708 Web Switch Port Usage

Port	Host	Layer 4 Processing
b1	Server A	Server
b2	Server B	Server
b6	Default Gateway Router. This connects the switch to the Internet where all client requests originate.	Client

   The ports are configured as follows:

```
>> Virtual server 1# /cfg/slb/port b1        (Select physical switch port b1)
>> SLB port B1# server ena                   (Enable server processing on port b1)
>> SLB port B1# ../port b2                   (Select physical switch port b2)
>> SLB port B2# server ena                   (Enable server processing on port b2)
>> SLB port B2# ../port b6                   (Select physical switch port b6)
>> SLB port B6# client ena                   (Enable client processing on port b6)
```

6. **On the California switch, enable SLB.**

```
>> SLB port B6# /cfg/slb                     (Select the SLB Menu)
>> Layer 4# on                               (Turn SLB on)
```

Step 3: Configure the California Site for GSLB

1. **On the California switch, define each remote site.**

   When you start configuring at the California site, California is local and Denver is remote. Add and enable the remote switch's IP address interface. In this example, there is only one remote site: Denver, with an IP interface address of 174.14.70.100. The following commands are used:

```
>> Server Load Balancing# gslb/site 1          (Select remote site #1)
>> Remote site 1# prima 174.14.70.100         (Define remote interface)
>> Remote site 1# ena                         (Enable remote site #1)
```

   Each additional remote site would be configured in the same manner. You can add and enable up to 128 remote sites.

2. **On the California switch, assign each remote distributed service to a local virtual server.**

NOTE – Take care to note where each configured value originates. If it is not clearly understood, this step can result in improper configuration.

   The local California site is configured to recognize the services offered at the remote Denver site. To do this, configure one real server entry on the California switch for each virtual server located at each remote site. Since there is only one remote site (Denver) with only one virtual server, only one more local real server entry is needed at the California site.

   The new real server entry is configured with the IP address of the remote virtual server rather than the usual IP address of a local physical server.

   Also, the *remote* property is enabled, and the real server entry is added to the real server group under the local virtual server for the intended service. Finally, since the real server health checks are performed across the Internet, the health-checking interval should be increased to 30 or 60 seconds to avoid generating excess traffic. For example:

```
>> Remote site 1# /cfg/slb/real 3             (Create an entry for real server #3)
>> Real server 3# rip 174.14.70.1             (Set remote virtual server IP address)
>> Real server 3# remote enable               (Define the real server as remote)
>> Real server 3# inter 60                    (Set a high health check interval)
>> Real server 3# ena                         (Enable the real server entry)
>> Real server 3# ../group 1                  (Select appropriate real server group)
>> Real server group 1# add 3                 (Add real server 3 to the group 1)
```

NOTE – The IP address of the real server being added is that of the virtual server IP address on the remote switch. Do not confuse this value with the *IP interface* address on the remote switch.

---

3. **On the California switch, define the domain name and host name for each service hosted on each virtual server.**

   In this example, the domain name for the Foo Corporation is "foocorp.com," and the host name for the only service (HTTP) is "www." These values are configured as follows:

```
>> Real server group 1# /cfg/slb/virt 1      (Select virtual server #1)
>> Virtual server 1# dname foocorp.com       (Define domain name)
>> Virtual server 1# service 80/hname www    (Define HTTP hostname)
```

   If other services were defined (such as FTP), additional host name entries would be required.

4. **On the California switch, turn on GSLB.**

```
>> Virtual server 1# ../gslb                 (Select the GSLB menu)
>> Global SLB# on                            (Activate GSLB for the switch)
```

5. **Apply and verify the configuration.**

```
>> Global SLB# apply                         (Make your changes active)
>> Global SLB# cur                           (View current GSLB settings)
>> Global SLB# /cfg/slb/cur                  (View current SLB settings)
```

   Examine the resulting information. If any settings are incorrect, make and apply any appropriate changes, and then check again.

6. **Save your new configuration changes.**

```
>> Layer 4# save                             (Save for restore after reboot)
```

Step 4: Configure the Basics at the Denver Site

Following the same procedure described for California (see "Step 1: Configure the Basics at the California Site" on page 213), configure the Denver site as follows:

1. **If the Web OS BBI is to be used for managing the Denver switch, change its service port.**

```
>> Main# /cfg/sys                            (Select the System Menu)
>> System# wport 8080                        (Set service port 8080 for WBI)
```

2. **On the Denver switch, define an IP interface.**

```
>> Main# /cfg/ip/if 1                          (Select IP interface #1)
>> IP Interface 1# addr 174.14.70.100          (Assign IP address for the interface)
>> IP Interface 1# ena                         (Enable IP interface #1)
```

NOTE – This example assumes that all ports and IP interfaces use default VLAN #1, requiring no special VLAN configuration for the ports or IP interface.

3. **On the Denver switch, define the default gateway.**

```
>> IP Interface 1# /cfg/ip/gw 1                (Select default gateway #1)
>> Default gateway 1# addr 174.14.70.101       (Assign IP address for the gateway)
>> Default gateway 1# ena                      (Enable default gateway #1)
```

4. **Configure the local DNS server to recognize the local GSLB switch as the authoritative name server for the hosted services.**

   The Denver DNS server is configured to recognize 174.14.70.100 (the IP interface of the Denver GSLB switch) as the authoritative name server for “www.foocorp.com.”

Step 5: Configure the Denver Switch for Standard SLB

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

Table 12-3 Denver Real Server IP Addresses

Real Server	IP address
Server C	174.14.70.2
Server D	174.14.70.3

2. **On the Denver switch, define each local real server.**

```
>> Default gateway 1# /cfg/slb/real 1          (Server C is real server 1)
>> Real server 1 # rip 174.14.70.2             (Assign Server C IP address)
>> Real server 1 # ena                         (Enable real server 1)
>> Real server 1 # ../real 2                   (Server D is real server 2)
>> Real server 2 # rip 174.14.70.3             (Assign Server D IP address)
>> Real server 2 # ena                         (Enable real server 2)
```

3. **On the Denver switch, define a real server group.**

```
>> Real server 2 # /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# health http          (Use HTTP for health checks)
>> Real server group 1# content index.html   (Set URL content for health checks)
```

4. **On the Denver switch, define a virtual server.**

```
>> Real server group 1 # ../virt 1           (Select virtual server 1)
>> Virtual server 1# vip 174.14.70.1         (Assign IP address)
>> Virtual server 1# service http            (Select the HTTP service menu)
>> Virtual server 1 http Service# group 1    (Associate virtual port to real group)
>> Virtual server 1 http Service# ../ena     (Enable the virtual server)
```

5. **On the Denver switch, set the type of Layer 4 traffic processing each port must support.**

In this example, the following ports are used on the Alteon 708 Web switch:

Table 12-4 Web Host Example: Alteon 708 Switch Port Usage

Port	Host	Layer 4 Processing
b3	Server C	Server
b4	Server D	Server
b5	Default Gateway Router. This connects the switch to the Internet where all client requests originate.	Client

The ports are configured as follows:

```
>> Virtual server 1# /cfg/slb/port b3        (Select physical switch port b3)
>> SLB port B3# server ena                   (Enable server processing on port b3)
>> SLB port B3# ../port b4                   (Select physical switch port b4)
>> SLB port B4# server ena                   (Enable server processing on port b4)
>> SLB port B4# ../port b5                   (Select physical switch port b5)
>> SLB port B5# client ena                   (Enable client processing on port b5)
```

6. **On the Denver switch, enable SLB.**

```
>> SLB port B5# /cfg/slb                     (Select the SLB menu)
>> Layer 4# on                               (Turn SLB on)
```

Step 6: Configure the Denver Site for GSLB

Configure the Denver site as follows which is similar to the procedure described for California (see "Step 3: Configure the California Site for GSLB" on page 216):

1. **On the Denver switch, define each remote site.**

   When you start configuring at the Denver site, Denver is local and California is remote. Add and enable the remote switch's IP address interface. In this example, there is only one remote site: California, with an IP interface address of 200.200.200.100. The following commands are used:

```
>> Main# cfg/slb/gslb/site 1                  (Select remote site 1)
>> Remote site 1# prima 200.200.200.100       (Define remote IP interface address)
>> Remote site 1# ena                         (Enable remote site 1)
```

   Each additional remote site would be configured in the same manner. You can add and enable up to 128 remote sites.

2. **On the Denver switch, assign each remote distributed service to a local virtual server.**

   NOTE – Take care to note where each configured value originates. If it is not clearly understood, this step can result in improper configuration.

   In this step, the local Denver site is configured to recognize the services offered at the remote California site. As before, configure one real server entry on the Denver switch for each virtual server located at each remote site. Since there is only one remote site (California) with only one virtual server, only one more local real server entry is needed at the Denver site.

   The new real server entry is configured with the IP address of the remote virtual server rather than the usual IP address of a local physical server.

   Also, the *remote* property is enabled, and the real server entry is added to the real server group under the local virtual server for the intended service. Finally, since the real server health checks are headed across the Internet, the health-checking interval should be increased to 30 or 60 seconds to avoid generating excess traffic. For example:

```
>> Remote site 1# /cfg/slb/real 3             (Create an entry for real server #3)
>> Real server 3# rip 200.200.200.1           (Set remote virtual server IP address)
>> Real server 3# remote enable               (Define the real server as remote)
>> Real server 3# inter 60                    (Set a high health check interval)
>> Real server 3# ena                         (Enable the real server entry)
>> Real server 3# ../group 1                  (Select appropriate real servr group)
>> Real server group 1# add 3                 (Add real server 3 to group 1)
```

NOTE – The IP address of the real server being added is taken from the virtual server IP address on the remote switch. Do not confuse this value with the *IP interface* address on the remote switch.

3. **On the Denver switch, define the domain name and host name for each service hosted on each virtual server.**

   These will be the same as for the California switch: the domain name is "foocorp.com," and the host name for the HTTP service is "www." These values are configured as follows:

```
>> Real server group 1# /cfg/slb/virt 1      (Select virtual server #1)
>> Virtual server 1# dname foocorp.com       (Define domain name)
>> Virtual server 1# service 80/hname www    (Define HTTP hostname)
```

4. **On the Denver switch, turn on GSLB.**

```
>> Virtual server 1# /cfg/slb/gslb/on        (Activate GSLB for the switch)
```

5. **Apply and verify the configuration.**

```
>> Global SLB# apply                         (Make your changes active)
>> Global SLB# cur                           (View current GSLB settings)
>> Global SLB# /cfg/slb/cur                  (View current GSLB settings)
```

   Examine the resulting information. If any settings are incorrect, make and apply any appropriate changes, and then check again.

6. **Save your new configuration changes.**

```
>> Layer 4# save                             (Save for restore after reboot)
```

Using Proxy IP for Non-HTTP Redirects

Web OS software allows you to configure GSLB remote servers to redirect client requests using a load-balancing mechanism called *IP proxy*. Under normal GSLB, client requests for HTTP applications are automatically redirected to the location with the best response and least load for the requested content. But if the client requests non-HTTP applications (FTP, POP3, or SMTP for example), then a proxy IP address is used on the client port which initiates a redirect only if resources are unavailable at the first site.

NOTE – This feature should be used as a method of last resort for GSLB implementations in topologies where the remote servers are usually virtual IP addresses in other Alteon WebSystems switches.

Figure 12-3 shows the path taken by HTTP redirects and non-HTTP redirects in a GSLB environment.

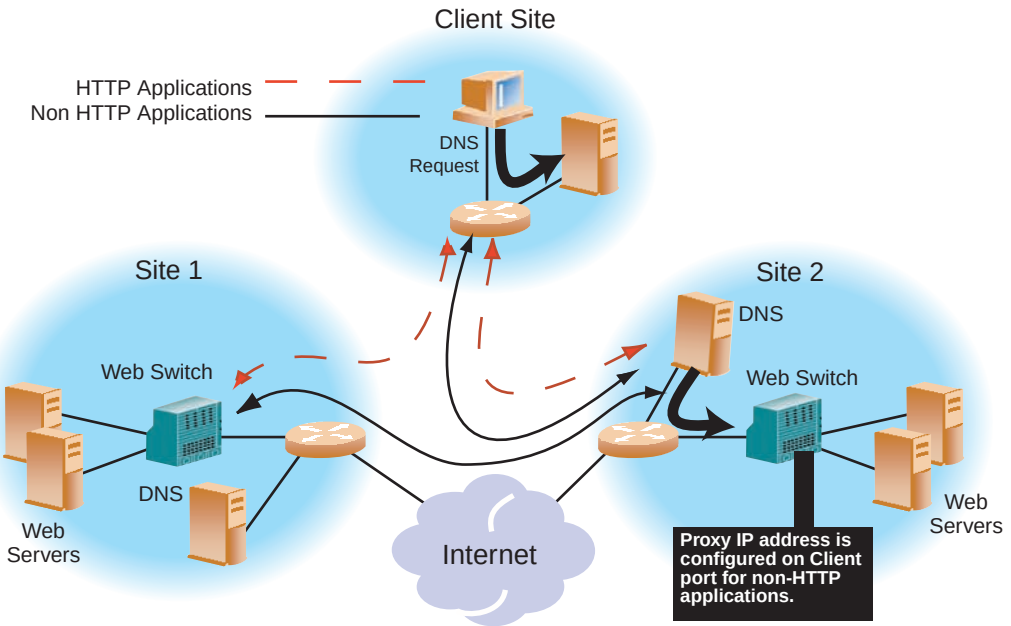


Figure 12-3 Using Proxy IP for Non-HTTP Redirects

Table 12-5 explains the process in the illustration in detail.The initial request from the Client reaches Site 2 because the local DNS is configured to forward requests to Site 2.

Table 12-5 HTTP versus Non-HTTP Redirects

	Web switch at Site 2	Web switch at Site 1
HTTP application	– Client request reaches Site 2 – Resources unavailable at Site 2 – Site 2 sends a response to Client with Site 1's virtual IP address.	– Client resends request to Site 1 – Resources available at Site 1 – Site 1 completes 3-way handshake with client
Non-HTTP application	– Client request reaches Site 2 – Resources unavailable at Site 2 – Site 2 sends a request to Site 1 with Site 2's proxy IP address as source IP address and virtual IP address of Site 1 as destination IP.	– Site 1 processes the client proxy IP request – Resources available at Site 1 – Site 1 returns request to proxy IP port on Site 2 – Site 2 completes the 3-way handshake with Client

How Proxy IP Works

Figure 12-4 shows two GSLB sites, each with one local virtual server serviced by two real servers in real server group 1. The applications being load balanced are HTTP and POP3. Any request that cannot be serviced locally is sent to the peer site. HTTP requests are sent to the peer site using HTTP Redirect. Any other application request will be sent to the peer site using the IP proxy feature.

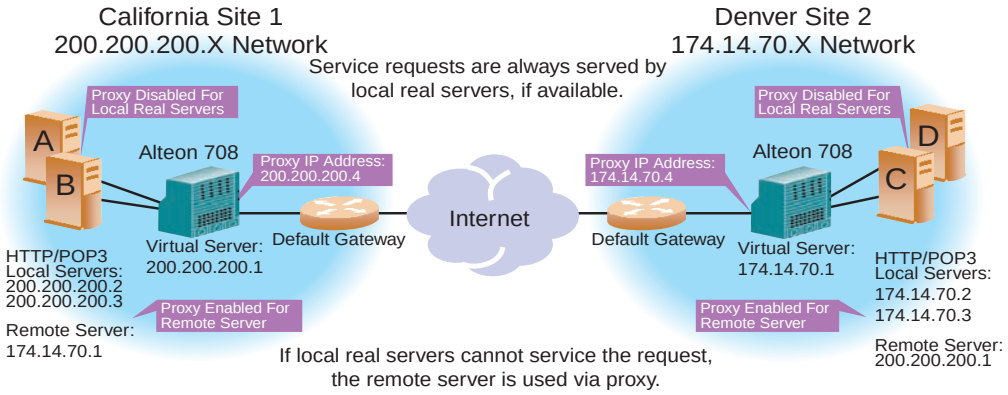


Figure 12-4 POP3 Request Fulfilled via IP Proxy

The following procedure explains the three-way handshake between the two sites and the client for a non-HTTP application (POP3). When POP3 processes at Site 1 terminate because of operator error, the following events occur to allow POP3 requests to be fulfilled.

1. **A user POP3 (TCP SYN) request is received by the virtual server at Site 1. The switch at that site determines that there are no local resources to handle the request.**
2. **Site 1 switch rewrites the request such that it now contains a client proxy IP address as the source IP address, and the virtual server IP address at Site 2 as the destination IP address.**
3. **The switch at Site 2 receives the POP3 (TCP SYN) request to its virtual server. The request looks like a normal SYN frame, so the switch performs normal local load-balancing.**
4. **Internally at Site 2, the switch and the real servers exchange information. The TCP SYN ACK from Site 2's local real server IP address is sent back to the IP address specified by the proxy IP address.**
5. **The Site 2 switch sends the TCP SYN ACK frame to Site 1, with Site 2's virtual server IP address as the sourceIP address, and Site 1's proxy IP address as the destination IP address.**
6. **The switch at Site 1 receives the frame and translates it, using Site 1's virtual server IP address as the source IP address and the client's IP address as the destination IP address.**

   This cycle continues for the remaining frames to transmit all the client's mail, until a `FIN` frame is received.

Configuring IP Proxy Address

In the example, Figure 12-2 on page 212, the switch at Site 1 in California is configured with switch port b6 (see Table 12-2) connecting to the default gateway and real server 3 representing the remote server in Denver. The following commands are used to configure the proxy IP address on Site 1 in California:

NOTE – Make sure you select the client port connecting to the default gateway.

```
>> # /cfg/slb/port b6                          (Select client port to default gateway)
>> SLB port B6# pip 200.200.200.4              (Set proxy IP address on the client port)
>> SLB port B6 ../real 1/proxy disable         (Disable local real server proxy)
>> Real server 1 # ../real 2/proxy disable     (Disable proxy for local server)
>> Real server 2 # ../real 3/proxy enable      (Enable proxy for remote server)
>> Real server 3 # apply                       (Apply configuration changes)
>> Real server 3 # save                        (Save configuration changes)
```

If you want to configure proxy IP addresses on Site 2, the following commands are issued on the Denver switch (see Table 12-4):

```
>> # /cfg/slb/port b5                          (Select client port to default gateway)
>> SLB port B5# pip 174.14.17.4                (Set proxy IP address on the client port)
>> SLB port B5# ../real 1/proxy disable        (Disable local real server proxy)
>> Real server 1 # ../real 2/proxy disable     (Disable local real server proxy)
>> Real server 2 # ../real 3/proxy enable      (Enable proxy for remote server)
>> Real server 3 # apply                       (Apply configuration changes)
>> Real server 3 # save                        (Save configuration changes)
```

Verifying GSLB Operation

- Use your browser to request the configured service (“www.foocorp.com” in the previous example).
- Examine the `/info/slb` information on each switch.
- Check that all SLB parameters are working according to expectation. If necessary, make any appropriate configuration changes and then check the information again.
- Examine the following statistics on each switch:
  - `/stats/slb/gslb/virt` *<virtual server number>*
  - `/stats/slb/gslb/group` *<real server group number>*
  - `/stats/slb/maint`

Configuring Client Site Preferences

Internet Assigned Numbers Authority (IANA), the central coordinator for the assignment of unique parameter values for internet protocols, does not provide sufficient geographic separation of client proximity information. As a result, large ISP partners cannot use their own geographic data to determine GSLB site selection based on client location. However, using static "client-to-site" mapping in Web OS software, you can configure client proximity tables to select GSLB sites based on client location.

The use of a static client/site database allows you to customize the client environment. The proximity database is limited to 128 entries. The GSLB client proximity table is supported for HTTP protocol only.

Figure 12-5 shows how GSLB proximity tables work. The clients sends a request to the DNS server, which is forwarded to the master switch. The master switch looks up its proximity table and returns the request to the DNS server with the virtual IP address of the preferred site. The DNS server sends out a new request through the Internet and connects the client to the preferred site.

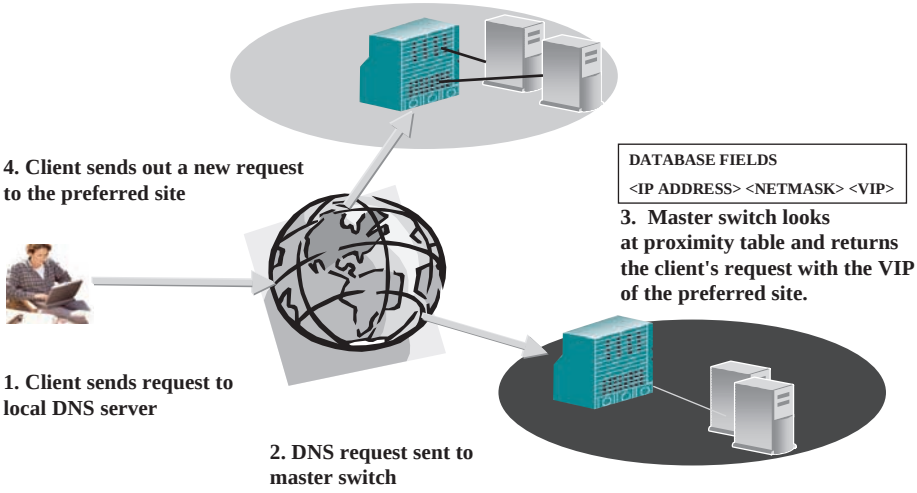


Figure 12-5 GSLB Proximity Tables: How They Work

The following example, illustrated in Figure 12-6, shows how to add entries into a GSLB proximity table. Two client networks, A and B, are configured in the proximity table on the master switch at Site 4. Client A, with a subnet address of 205.178.13.0, is configured with static entries for preferred Sites 1 and 3. Client B, with a subnet address of 204.165.0.0, is configured with static entries for preferred Sites 2 and 4.

Client A, with a source IP address 205.178.13.10, initiates a request that is sent to the local DNS server. The local DNS server is configured to forward requests to the DNS server at site 4. The Alteon 708 at site 4 looks up its proximity table and finds Client A prefers to connect to sites 1 or 3. Similarly, Client B's requests are always forwarded to Sites 2 or 4.

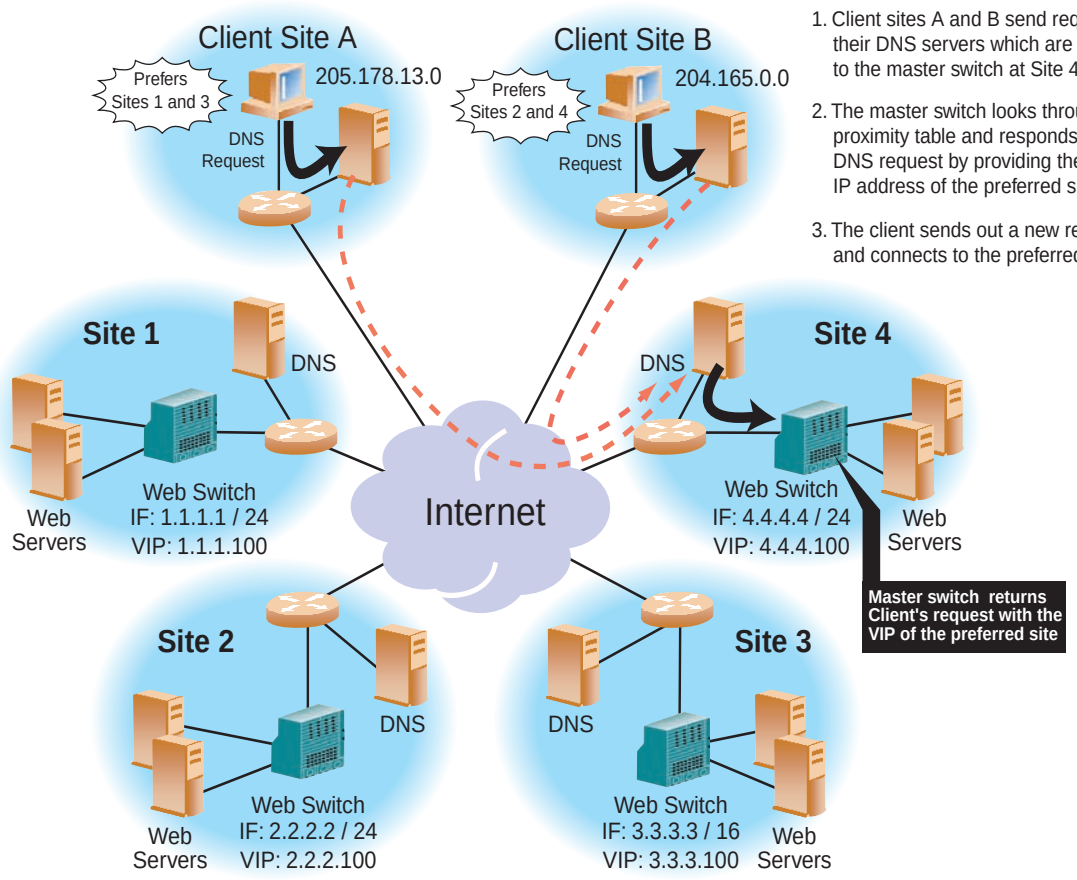


Figure 12-6 Configuring Client Proximity Table

You can add static entries in the proximity table for up to 128 client networks.

NOTE – Web OS allows you to configure a single domain only; for each client subnet, you can configure up to two preferred sites.

The Web switch forwards the client request based on the minimum available sessions and the response time between the two preferred sites.

Use the following commands to configure the proximity table on the Alteon 708 at Site 4 in Figure 12-6:

```
>> # /cfg/slb/gslb/lookup/lookups ena     (Enable the lookup or proximity table)
>> # dname nortelnetworks.com             (Select the domain name)
>> # network 1                            (Select Client A subnet)
>> # sip 205.178.13.0                     (Assign the SIP for Client A)
>> # mask 255.255.255.0                   (Assign the mask for Client A)
>> # vip1 3.3.3.100                       (Select preferred Site 3)
>> # vip2 1.1.1.100                       (Select preferred Site 1)
>> # ../network 2                         (Select Client B subnet)
>> # sip 204.165.0.0                      (Assign the SIP for Client B)
>> # mask 255.255.0.0                     (Assign the mask for Client B)
>> # vip1 4.4.4.100                       (Select preferred Site 4)
>> # vip2 2.2.2.100                       (Select preferred Site 2)
```

NOTE – For each client subnet, add only one static entry.

Using this configuration, the DNS request `nortelnetworks.com` from 205.178.13.0 will get a DNS response with only the virtual server IP address of Site 1, that is, if Site 1 has less load than Site 3.

Using Border Gateway Protocol for GSLB

Border Gateway Protocol (BGP)-based Global Server Load Balancing utilizes the Internet's routing protocols to localize content delivery to the most efficient and consistent site. It does this through the use of a shared IP block that co-exists in each ISP's network and which is then advertised, via BGP, throughout the Internet. Because of the way IP routing works, BGP-based GSLB allows for the routing protocols to route DNS requests to the closest location, which then returns IP addresses of that particular site, locking in the requests to that site. In effect, the Internet is making the decision of the "best" location for you, avoiding the need for advanced GSLB.

For more information on configuring your switch to support BGP routing, see "Border Gateway Protocol" on page 75.

Instead of multi-homing a network to several other networks, BGP-based GSLB enables you to multi-home a particular piece of content (DNS information) to the Internet by distributing the IP blocks that contain that content to several sites.

When using DNS to select the site, a single packet is used to make the decision and the request is not split to two different locations. Through the response to the DNS packet, a client is locked down to a particular site, resulting in efficient, consistent service that cannot be achieved when the content itself is shared.

For example, in multi-homing, you can connect a data center to three different Internet providers, and have one DNS server that has the IP address 1.1.1.1. In this case, all requests are received in via any given feed but are funneled to the same server on the local network. In BGP-based GSLB, the DNS server (with the IP address 1.1.1.1) is simply duplicated and placed local to the *peering point* instead of having a local network direct traffic to one server. When a particular DNS server receives a request for a record (in this case, the Alteon switch), it returns with the IP address of a virtual server at the same site. This can be achieved using the `local` option (`/cfg/slb/gslb/local ena`) in the GSLB configuration. If one site is saturated, the switch will failover and deliver the IP address of a more available site.

CHAPTER 13

Firewall Load Balancing

Network security has become an increasing concern for Web hosters and Internet Service Providers (ISPs). As a result, firewalls are commonly used to prevent unauthorized access to network resources. While firewalls are very effective at preventing network intrusions, they are often single points-of-failure and can reduce network availability. The following topics are discussed in this chapter:

- "Overview" on page 232. This section explains the concept of firewall load balancing and a basic configuration.
- "Adding a DMZ" on page 242. This section describes how the Web switch off-loads tasks from the firewall by using a De-Militarized Zone (DMZ) in conjunction with firewall load balancing.
- "Firewall Health Checks" on page 244. Each Web switch monitors the health of the firewalls on a regular basis by pinging the IP interfaces configured on its partner Web switch on the other side of the firewall.
- "FWLB with High Availability Checklist" on page 246. This section provides a checklist for configuring VRRP on the primary and secondary Web switches.
- "Firewall Configuration Examples" on page 247. This section describes four different configuration examples for configuring firewall load balancing.

Firewall Load Balancing (FWLB) with Alteon Web switches divides the load between many discrete firewalls, allowing multiple active firewalls to operate in parallel. Parallel operation allows users to maximize firewall productivity, scale firewall performance without forklift upgrades, and eliminate the firewall as a single point-of-failure.

Overview

Typically, a firewall server is inserted into the data path between the private (protected) network and the public network.

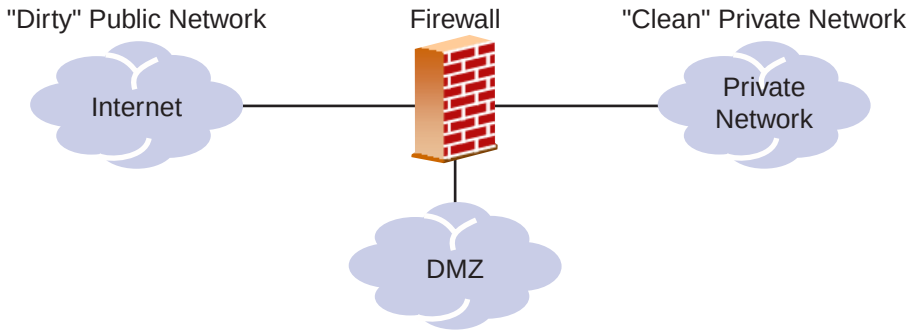


Figure 13-1 Typical Firewall Configuration Before FWLB

One network interface card on the firewall is connected to the public side of the network, often to an Internet router. This is known as the *dirty* or untrusted side of the firewall.

Another network interface card on the firewall is connected to the side of the network with the resources that must be protected. This is known as the *clean* or trusted side of the firewall.

All network traffic passing between the dirty and clean networks must traverse the firewall, which examines each individual packet. With firewall load balancing, filters that redirect all incoming IP traffic are configured on Web switches on both the dirty and clean sides of the firewall. On each switch, the filtered traffic is redirected in such a way as to load balance it across the firewalls. Because the switches intelligently maintain state information about the traffic flowing through them, they ensure that all traffic between specific IP source/destination address pairs flows through the same firewall. This, in turn, ensures that sessions established by the firewalls are maintained for their duration.

NOTE – While you can implement single-switch firewall load balancing, the switch will perform health checking on the firewall interfaces only.

Basic FWLB Implementation

The following figure shows a basic FWLB topology. This configuration requires a minimum of two Web switches; for extra redundancy, four switches and hubs can be used.

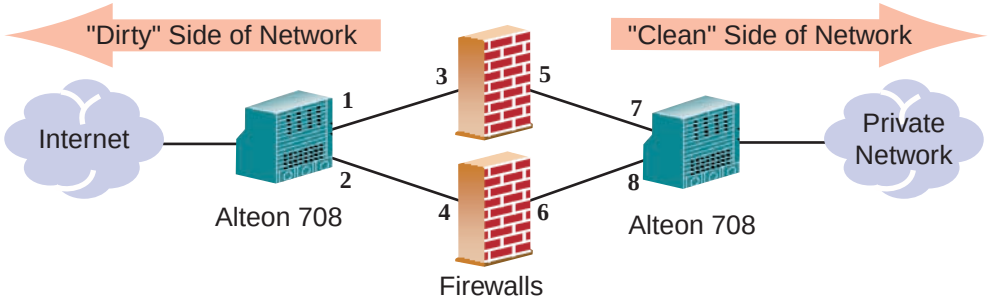


Figure 13-2 Basic FWLB Topology

The two firewalls being load balanced are in the middle of the network, separating the dirty side from the clean side. Two Web switches are also in place, one on the clean side of the firewalls and one on the dirty side.

NOTE – FWLB generally requires VLANs. If you will be using hubs and do not configure VLANs, you must enable Spanning Tree.

In the following two-switch configuration example, external clients connect to services at an IP address assigned to a virtual server on the clean-side Web switch. With Alteon WebSystems FWLB, data inbound or outbound to/from the Internet is load balanced among the available firewalls. Here's how the process works:

1. **The client request arrives at this network via the ingress port on the dirty-side Web switch.**
2. **A redirection filter balances incoming requests between two IP addresses.**

   A filter is configured on the ingress port which will redirect requests to a real server group that consists of two different IP addresses. Each IP address represents an IP interface on a different subnet on the clean-side switch on the other side of the firewall.

3. **Requests are routed to the firewalls.**

   Two static routes are configured on the dirty-side switch. The first static route leads to one of the clean-side switch IP interfaces using one firewall as the next hop. The second static route leads to the other clean-side IP interface using the other firewall as the next hop. By combining the redirection filter and static routes, the client requests are load balanced between both firewalls.

Because Web switches intelligently maintain state information about the traffic flowing through them, they ensure that all traffic between specific IP source/destination address pairs flows through the same firewall. This ensures that sessions established by the firewalls are maintained for their duration.

4. **The firewall decides if it should allow the packet and, if so, where to forward it.**

   Client requests are forwarded or discarded according to rule sets configured for each firewall.

NOTE – Rule sets must be consistent across all firewalls.

   Forwarded packets are sent on to the original destination address: That is, the virtual server on the clean-side switch, where they are load balanced to the real servers using standard (SLB) configuration.

5. **The server response arrives at the ingress port on the clean-side Web switch.**

6. **Redirection filters balance responses between two IP addresses.**

   Redirection filters are needed on all ingress ports on the clean-side Web switch that attach to real servers or internal clients on the clean-side of the network. Filters on these ports redirect the outbound traffic to a real server group on the dirty-side switch that consists of two different IP addresses. Each IP address represents an IP interface on two more subnets on the dirty-side switch on the other side of the firewall.

   Web switches intelligently maintain state information. If Network Address Translation (NAT) is not being used, all traffic between specific IP source/destination address pairs flows through the same firewall.

7. **Outbound traffic is routed to the firewalls.**

   Two static routes are configured on the clean-side switch. The first static route leads to one of the dirty-side IP interfaces, using one firewall as the next hop. The second static route leads to the other dirty-side IP interface, using the other firewall as the next hop.

8. **The firewall decides if it should allow the packet and, if so, where to forward it.**

   Each firewall forwards or discards the client requests according to the rules that are configured for it. Forwarded packets are sent on to the dirty-side Web switch, and from there, out to their destination on the Internet.

Configuring Basic FWLB

The steps for configuring basic firewall load balancing are provided below. While two or four switches can be used, the following procedure assumes a network topology having one Web switch on either side of the firewalls.

Table 13-1 lists the IP addresses for the interfaces shown in Figure 13-2.

Table 13-1 IP Interface and IP addresses

Network device	IP interface	IP address
Alteon 708	1	192.168.13.1/24
Alteon 708	2	192.168.16.1/24
Firewall	3	192.168.13.2/24
Firewall	4	192.168.16.2/24
Firewall	5	192.168.14.2/24
Firewall	6	192.168.18.2/24
Alteon 708	7	192.168.14.1/24
Alteon 708	8	192.168.18.1/24

Configure the Dirty-Side Web Switch

1. **Define the dirty-side IP interfaces.**

   There should be one dirty-side IP interface on a different subnet for each firewall being load balanced.

```
>> # /cfg/ip/if 1                          (Select IP interface 1)
>> IP Interface 1# addr 192.168.13.1       (Set the IP address for interface 1)
>> IP Interface 1# mask 255.255.255.0      (Set subnet mask for interface 1)
>> IP Interface 1# ena                     (Enable IP interface 1)
>> IP Interface 1# ../if 2                 (Select IP interface 2)
>> IP Interface 2# addr 192.168.16.1       (Set the IP address for interface 2)
>> IP Interface 2# mask 255.255.255.0      (Set subnet mask for interface 2)
>> IP Interface 2# ena                     (Enable IP interface 2)
```

2. **Configure real servers using the IP addresses of the clean-side IP interfaces.**

   Later in this procedure, you'll configure one clean-side IP interface on a different subnet for each firewall being load balanced. Create two real servers on the dirty-side switch, using the IP address of each clean-side IP interface.

```
>> IP Interface 2# /cfg/slb/real 1          (Select real server 1)
>> Real server 1# rip 192.168.14.1          (Assign clean-side IF 3 address)
>> Real server 1# ena                       (Enable real server 1)
>> Real server 1# ../real 2                 (Select real server 2)
>> Real server 2# rip 192.168.18.1          (Assign clean-side IF 4 address)
>> Real server 2# ena                       (Enable real server 1)
```

NOTE – Each of the four firewall interfaces (two on each Web switch) in this example must be configured for a different IP subnet.

3. **Place the real servers into a real server group.**

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

4. **Set the health check type for the real server group to ICMP.**

```
>> Real server group 1# health icmp         (Select ICMP as health check type)
```

5. **Set the load-balancing metric for the real server group to `hash`.**

```
>> Real server group 1# metric hash         (Select SLB hash metric for group 1)
```

6. **Enable SLB on the switch.**

```
>> Real server group 1# /cfg/slb/on
```

7. **Create the filters to allow local subnet traffic on the dirty side of the firewalls to reach the firewall interfaces.**

   Two filters are needed to prevent local traffic from being redirected to firewalls.

   NOTE – Local traffic allowed includes VRRP updates and traffic between specific devices on the dirty side of the network.

```
>> Layer 4# /cfg/slb/filt 110          (Select filter 110)
>> Filter 110# sip any                 (From any source IP address)
>> Filter 110# dip 192.168.13.0        (To this destination IP address)
>> Filter 110# action allow            (Allow frames with this DIP address)
>> Filter 110# ena                     (Enable filter)
>> Filter 110# ../filt 120             (Select filter 120)
>> Filter 120# sip any                 (From any source IP address)
>> Filter 120# dip 192.168.16.0        (To this destination IP address)
>> Filter 120# action allow            (Allow the traffic)
>> Filter 120# ena                     (Enable filter)
```

8. **Create the redirection filter.**

   This filter will redirect inbound traffic, load balancing it among the defined real servers in the group. In this network, the real servers represent IP interfaces on a clean-side switch.

```
>> Filter 120# ../filt 1023            (Select filter 1023)
>> Filter 1023# sip any                (From any source IP address)
>> Filter 1023# dip any                (To any destination IP address)
>> Filter 1023# proto any              (For any protocol)
>> Filter 1023# action redir           (Perform redirection)
>> Filter 1023# group 1                (To real server group 1)
>> Filter 1023# ena                    (Enable the filter)
```

9. **Add filters to the ingress port.**

```
>> Filter 1023# ../port a5             (Select the ingress port)
>> SLB Port A5# add 110                (Add the filter to the ingress port)
>> SLB Port A5# add 120                (Add the filter to the ingress port)
>> SLB Port A5# add 1023               (Add the filter to the ingress port)
>> SLB Port A5# filt ena               (Enable filtering on the port)
```

10. **Define static routes to the clean-side IP interfaces, using the firewalls as gateways.**

    One static route is required for each firewall being load balanced. In this case, two paths are required: one that leads to clean-side IP interface 3 (192.168.14.1) through the first firewall (192.168.13.2) as its gateway, and one that leads to clean-side IP interface 4 (192.168.18.1) through the second firewall (192.168.16.2) as its gateway.

```
>> SLB Port A5# /cfg/ip/route
>> IP Static Route# add 192.168.14.1 255.255.255.255 192.168.13.2 1
>> IP Static Route# add 192.168.18.1 255.255.255.255 192.168.16.2 2
```

11. **Apply and save the configuration changes.**

```
>> # apply
>> # save
```

Configure the Clean-Side Web Switch

1. **Define the clean-side IP interfaces**

   Create one clean-side IP interface on a different subnet for each firewall being load balanced.

NOTE – An extra IP interface (IF 5) prevents server-to-server traffic from being redirected.

```
>> Main# /cfg/ip/if 7                      (Select IP interface 7)
>> IP Interface 7# addr 192.168.14.1       (Set the IP address for interface 7)
>> IP Interface 7# mask 255.255.255.0      (Set subnet mask for interface 7)
>> IP Interface 7# ena                     (Enable IP interface 7)
>> IP Interface 7# ../if 8                 (Select IP interface 8)
>> IP Interface 8# addr 192.168.18.1       (Set the IP address for interface 8)
>> IP Interface 8# mask 255.255.255.0      (Set subnet mask for interface 8)
>> IP Interface 8# ena                     (Enable IP interface 8)
>> IP Interface 8# ../if 9                 (Select IP interface 9)
>> IP Interface 9# addr 192.168.20.1       (Set the IP address for interface 9)
>> IP Interface 9# mask 255.255.255.0      (Set subnet mask for interface 9)
>> IP Interface 9# ena                     (Enable IP interface 9)
```

2. **Configure real servers that have IP addresses of the dirty-side IP interfaces.**

   You should already have configured a dirty-side IP interface on a different subnet for each firewall being load balanced. Create two real servers on the clean-side switch, using the IP address of each dirty-side IP interface.

```
>> #/cfg/slb/real 1                    (Select real server 1)
>> Real server 1 # rip 192.168.13.1    (Assign dirty-side IF 1 address)
>> Real server 1 # ena                 (Enable real server 1)
>> Real server 1 # ../real 2           (Select real server 2)
>> Real server 2 # rip 192.168.16.1    (Assign dirty-side IF 2 address)
>> Real server 2 # ena                 (Enable real server 2)
```

NOTE – Each of the four IP interfaces (two on each Web switch) in this example must be configured for a different IP subnet.

3. **Place the real servers into a real server group.**

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

4. **Set the health check type for the real server group to ICMP.**

```
>> Real server group 1# health icmp    (Select ICMP as health check type)
```

5. **Set the load-balancing metric for the real server group to `hash`.**

```
>> Real server group 1# metric hash    (Select SLB hash metric for group 1)
```

6. **Enable server load balancing on the switch.**

```
>> Real server group 1# /cfg/slb/on
```

7. **Create a filter to prevent server-to-server traffic from being redirected.**

```
>> Layer 4# /cfg/slb/filt 100                (Select filter 100)
>> Filter 100# sip any                       (From any source IP address)
>> Filter 100# dip 192.168.20.0              (To base IP address for IF 5)
>> Filter 100# dmask 255.255.255.0           (For the range of addresses)
>> Filter 100# proto any                     (For any protocol)
>> Filter 100# action allow                  (Allow traffic)
>> Filter 100# ena                           (Enable the filter)
```

8. **Create the filters to allow local subnet traffic.**

Two filters are needed to prevent local traffic from being redirected to firewalls.

```
>> Layer 4# ../filt 110                      (Select filter 110)
>> Filter 110# sip any                       (From any source IP address)
>> Filter 110# dip 192.168.14.0              (To this destination IP address)
>> Filter 110# action allow                  (Allow frames with this DIP address)
>> Filter 110# ena                           (Enable filter)
>> Filter 110# ../filt 120                   (Select filter 120)
>> Filter 120# sip any                       (From any source IP address)
>> Filter 120# dip 192.168.18.0              (To this destination IP address)
>> Filter 120# action allow                  (Allow the traffic)
>> Filter 120# ena                           (Enable filter)
```

9. **Create the redirection filter.**

This filter will redirect outbound traffic, load balancing it among the defined real servers in the group. In this case, the real servers represent IP interfaces on the dirty-side switch.

```
>> Filter 120# ../filt 1023                  (Select filter 1023)
>> Filter 1023# sip any                      (From any source IP address)
>> Filter 1023# dip any                      (To any destination IP address)
>> Filter 1023# proto any                    (For any protocol)
>> Filter 1023# action redir                 (Perform redirection)
>> Filter 1023# group 1                      (To real server group 1)
>> Filter 1023# ena                          (Enable the filter)
```

10. Add the filters to the ingress ports for the outbound packets.

Redirection filters are needed on all the ingress ports on the clean-side Web switch. Ingress ports are any that attach to real servers or internal clients on the clean-side of the network. In this case, two real servers are attached to the clean-side Web switch on port A2 and port A6.

```
>> Filter 1023# ../port a2          (Select ingress port a2)
>> SLB Port A2# add 100             (Add the filter to the ingress port)
>> SLB Port A2# add 110             (Add the filter to the ingress port)
>> SLB Port A2# add 120             (Add the filter to the ingress port)
>> SLB Port A2# add 1023            (Add the filter to the ingress port)
>> SLB Port A2# filt ena            (Enable filtering on the port)
>> SLB Port A2# ../port a6          (Select ingress port a6)
>> SLB Port A6# add 100             (Add the filter to the ingress port)
>> SLB Port A6# add 110             (Add the filter to the ingress port)
>> SLB Port A6# add 120             (Add the filter to the ingress port)
>> SLB Port A6# add 1023            (Add the filter to the ingress port)
>> SLB Port A6# filt ena            (Enable filtering on the port)
```

11. Define static routes to the dirty-side IP interfaces, using the firewalls as gateways.

One static route is required for each firewall being load balanced. In this case, two paths are required: one that leads to dirty-side IP interface 1 (192.168.13.1) through the first firewall (192.168.14.2) as its gateway, and one that leads to dirty-side IP interface 2 (192.168.16.1) through the second firewall (192.168.18.2) as its gateway.

```
>> SLB Port A5# /cfg/ip/route
>> IP Static Route# add 192.168.13.1 255.255.255.255 192.168.14.2 7
>> IP Static Route# add 192.168.16.1 255.255.255.255 192.168.18.2 8
```

NOTE – Configuring static routes for FWLB does not require IP forwarding to be turned on.

12. Apply and save the configuration changes.

```
>> # apply
>> # save
```

Adding a DMZ

Using a De-Militarized Zone (DMZ) in conjunction with firewall load balancing, the Web switch filters the traffic, off-loading this task from the firewall. A DMZ is created by configuring firewall load balancing on the same switch with another real server group and a redirection filter towards the DMZ subnets.

The DMZ servers can be connected to the Web switch on the dirty side of the firewall. A typical firewall load balancing configuration with a DMZ is shown in the following illustration.

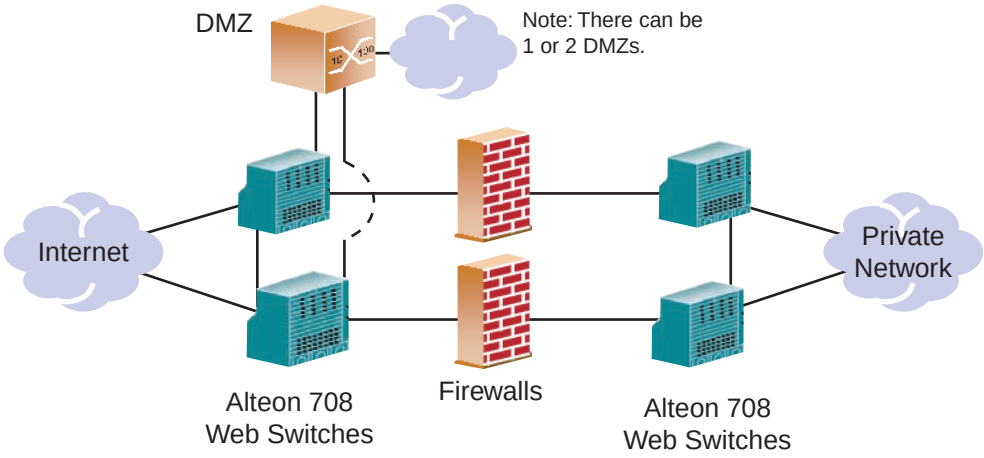


Figure 13-3 Typical Firewall Load-Balancing Topology with DMZ

The DMZ servers can be attached to the Web switch directly or through an intermediate hub or switch. The Web switch is then configured with filters to permit or deny access to the DMZ servers. In this manner, two levels of security are implemented: one that restricts access to the DMZ through the use of Web switch filters, and another that restricts access to the clean network through the use of stateful inspection performed by the firewalls.

You could add the filters required for the DMZ as follows:

1. **On the dirty-side Web switch, create the filter to allow HTTP traffic to the DMZ Web servers.**

```
>> #/cfg/slb/filt 80                    (Select filter 80)
>> Filter 80# sip any                   (From any source IP address)
>> Filter 80# dip 205.178.29.0          (To the DMZ base destination)
>> Filter 80# dmask 255.255.255.0       (For the range of DMZ addresses)
>> Filter 80# proto tcp                 (For TCP protocol traffic)
>> Filter 80# sport any                 (From any source port)
>> Filter 80# dport http                (To an HTTP destination port)
>> Filter 80# action allow              (Allow the traffic)
>> Filter 80# ena                       (Enable the filter)
```

2. **Create another filter to deny all other traffic to the DMZ Web servers.**

```
>> Filter 80# ../filt 89                (Select filter 89)
>> Filter 89# sip any                   (From any source IP address)
>> Filter 89# dip 205.178.29.0          (To the DMZ base destination)
>> Filter 89# dmask 255.255.255.0       (For the range of DMZ addresses)
>> Filter 89# proto any                 (For TCP protocol traffic)
>> Filter 89# action deny               (Allow the traffic)
>> Filter 89# ena                       (Enable the filter)
```

NOTE – The `deny` filter has a higher filter number than the `allow` filter. This is necessary so that the `allow` filter has the higher order of precedence.

---

3. **Add the filters to the traffic ingress ports.**

```
>> Filter 89# ../port a5                (Select the ingress port)
>> SLB Port A5# add 80                  (Add the allow filter)
>> SLB Port A5# add 89                  (Add the deny filter)
```

4. **Apply and save the configuration changes.**

```
>> SLB Port A5# apply
>> SLB Port A5# save
```

Firewall Health Checks

Basic FWLB health checking, as discussed below, is automatic. No special configuration is necessary unless you wish to tune the health checking parameters. See Chapter 10, "Health Checking" for details.

Firewall Service Monitoring

To maintain high availability, Web switches monitor firewall health status and send packets only to healthy firewalls. There are two methods of firewall service monitoring: ICMP and HTTP. Each Web switch monitors the health of the firewalls on a regular basis by pinging the IP interfaces configured on its partner Web switch on the other side of the firewall.

If a Web switch interface fails to respond to a user-specified number of pings, it (and, by implication, the associated firewall), is placed in a *server failed* state. At this time, the partner Web switch stops routing traffic to that interface, instead distributing it across the remaining, healthy Web switch interfaces and firewalls.

When a Web switch interface is in the *server failed* state, its partner Web switch continues to send pings to it at a user-configurable rate. After a specified number of successful pings, the interface (and its associated firewall) is brought back into service.

NOTE – To configure the switch to allow one-second intervals between health checks/pings, two failed health checks to remove the firewall, and four successful health checks to restore the firewall to the real server group, use the following command: `/cfg/slb/real` *x*`/inter 1/retry 2/restr 4`.

Physical Link Monitoring

Web switches also monitor physical link status of switch ports connected to firewalls. If the physical link to a firewall goes down, that firewall is placed immediately in the *server failed* state. When a Web switch detects that a failed physical link to a firewall has been restored, it brings the firewall back into service.

Using HTTP Health Checks

For those firewalls that do not permit ICMP pings to pass through, Web switches can be configured to perform HTTP health checks, as described below.

1. **Set the health check type to HTTP instead of ICMP.**

```
>> #/cfg/slb/group 1/health http          (Select HTTP health checks)
```

2. **Configure a "dummy" redirect filter as the last filter (after the "redirect all" filter) to force the HTTP health checks to activate, as shown below:**

```
>> #/cfg/slb/filt 1024                    (Select filter 1024)
>> Filter 1024# proto tcp                 (For TCP protocol traffic)
>> Filter 1024# action redir              (Redirect the traffic)
>> Filter 1024# group 1                   (Set real server group for redirection)
>> Filter 1024# rport http                (Set real server port for redirection)
>> Filter 1024# ena                       (Enable the filter)
```

NOTE – Make sure that the number of each real filter is lower than the number of the "dummy" redirect filter.

FWLB with High Availability Checklist

NOTE – *Before* you configure any switches, make sure that the firewalls work as firewalls.

Primary Switch

- Create the switch interfaces. Add the data interface first. If needed, add them to VLANs.
- Add the real servers. Make appropriate ones backup, if needed. Remember, backup real servers must be in order. For example, if the real server is 1, then its backup would be real server 2.
- Add the real servers to a group.
- Set load balancing metric to `hash`.
- Set the `health` option to ICMP (or HTTP).
- Turn on SLB.
- Turn on VRRP. VRRP is required in a redundant switch configuration only as shown in Figure 13-4.
- Create VRRP addresses.
- Create the static routes to the opposite switch's interfaces.
- Create the filters for VRRP multicast, local traffic `allow`, and everything else `redir`.
- Add filters to ingress port.
- Turn IP Forwarding on = VLAN configuration.
  Turn IP Forwarding off = Single VLAN approach.
- Turn off (or tune) Spanning Tree.

Secondary Switch

- Create the switch interfaces.
- Turn on SLB.
- Copy the static routes from the primary switch.
- Turn on VRRP.
- Synchonize from primary (`/oper/slb/s`) IP of secondary, execute from primary.

Firewall Configuration Examples

- Four Subnet Firewall Load Balancing using Alteon 708 Web switches
- Four Switches, Two Routing Firewalls, with No Interior Hubs on page 253
- Configuring NAT on Solaris Firewalls on page 260
- NAT for Single Devices on the Clean-Side Network on page 261

Four Subnet FWLB Using Alteon 708 Web Switches

This example shows how to set-up a four-switch firewall load-balancing sandwich, using Alteon 708 switches for the dirty side and on the clean side of the network and two firewalls. Figure Figure 13-4 shows the desired topology.

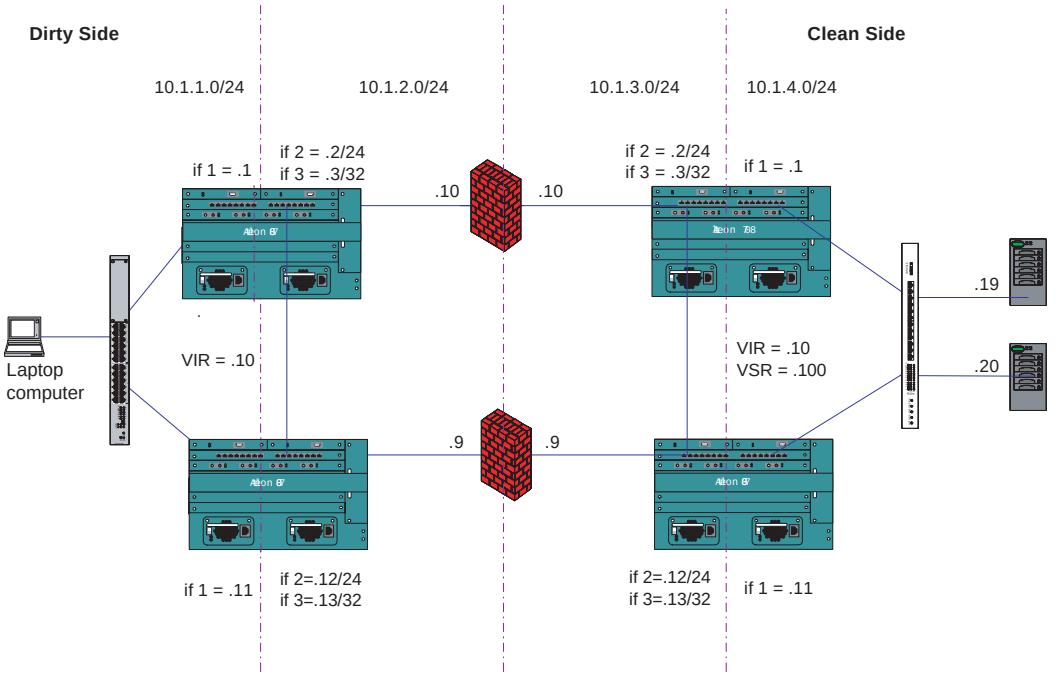


Figure 13-4 Four Subnet FWLB with Alteon 708 Web Switches

The procedure is divided into the following tasks:

- Set up VLANs, IP addresses, and static routes (with interface numbers) on all devices.
- Verify connectivity.
- On both the clean and dirty switches, create a group of real servers using the far switches' IP interfaces as real server IP addresses and set the metric for the group to `hash`.
- Create allow filters for VRRP and local traffic.
- Create a redirect filter for all traffic that needs to pass through the firewalls.
- Set up SLB on the Alteon 708s with two real servers in a group, and assign the server groups to a virtual server.

NOTE – Before configuring firewall load balancing, ensure the two firewalls have been set up for normal Layer 2/3 routing.

---

1. **Configure Layer 2/3 routing on the primary dirty-side Web switch.**

NOTE – When defining static routes, it is important to specify interface numbers.

```
>> #/cfg/vlan 2/add a7/add a8/ena

>> #/cfg/ip/if 1/mask 255.255.255.0/addr 10.1.1.1/ena
>> #/cfg/ip/if 2/vlan 2/mask 255.255.255.0/addr 10.1.2.2/ena
>> #/cfg/ip/if 3/vlan 2/mask 255.255.255.255/addr 10.1.2.3/ena

>> #/cfg/ip/frwd/on

>> #/cfg/ip/route
>> #add 10.1.3.2 255.255.255.255 10.1.2.10 2
>> #add 10.1.3.12 255.255.255.255 10.1.2.9 2
>> #add 10.1.3.3 255.255.255.255 10.1.2.10 3
>> #add 10.1.3.13 255.255.255.255 10.1.2.9 3

>> #/cfg/stp 2/off
>> #apply
>> #save
>> #/boot/reset
```

2. **Configure Layer 2/3 routing on the secondary dirty-side switch.**

```
>> #/cfg/vlan 2/add a7/add a8/ena

>> #/cfg/ip/if 1/mask 255.255.255.0/addr 10.1.1.11/ena
>> #/cfg/ip/if 2/vlan 2/mask 255.255.255.0/addr 10.1.2.12/ena
>> #/cfg/ip/if 3/vlan 2/mask 255.255.255.255/addr 10.1.2.13/ena
>> #/cfg/ip/frwd/on

>> #/cfg/ip/route
>> #add 10.1.3.2 255.255.255.255 10.1.2.10 2
>> #add 10.1.3.12 255.255.255.255 10.1.2.9 2
>> #add 10.1.3.3 255.255.255.255 10.1.2.10 3
>> #add 10.1.3.13 255.255.255.255 10.1.2.9 3

>> #/cfg/stp 2/off
>> #apply
>> #save
>> #/boot/reset
```

3. **Configure Layer 2/3 routing on the primary clean-sideWeb switch.**

```
>> #/cfg/vlan 2/add A1/add A2/ena

>> #/cfg/ip/if 1/mask 255.255.255.0/addr 10.1.4.1/ena
>> #/cfg/ip/if 2/vlan 2/mask 255.255.255.0/addr 10.1.3.2/ena
>> #/cfg/ip/if 3/vlan 2/mask 255.255.255.255/addr 10.1.3.3/ena

>> #/cfg/ip/frwd/on

>> #/cfg/ip/route
>> #add 10.1.2.2 255.255.255.255 10.1.3.10 2
>> #add 10.1.2.12 255.255.255.255 10.1.3.9 2
>> #add 10.1.2.3 255.255.255.255 10.1.3.10 3
>> #add 10.1.2.13 255.255.255.255 10.1.3.9 3

>> #/cfg/stp 1/off
>> #apply
>> #save
>> #/boot/reset
```

4. **Configure Layer 2/3 routing on the secondary clean-side switch (Alteon 708 switch).**

```
>> #/cfg/vlan 2/add A1/add A2/ena

>> #/cfg/ip/if 1/mask 255.255.255.0/addr 10.1.4.11/ena
>> #/cfg/ip/if 2/vlan 2/mask 255.255.255.0/addr 10.1.3.12/ena
>> #/cfg/ip/if 3/vlan 2/mask 255.255.255.255/addr 10.1.3.13/ena

>> #/cfg/ip/frwd/on

>> #/cfg/ip/route
>> #add 10.1.2.2 255.255.255.255 10.1.3.10 2
>> #add 10.1.2.12 255.255.255.255 10.1.3.9 2
>> #add 10.1.2.3 255.255.255.255 10.1.3.10 3
>> #add 10.1.2.13 255.255.255.255 10.1.3.9 3

>> #/cfg/stp 1/off
>> #apply
>> #save
>> #/boot/reset
```

5. **Verify connectivity by pinging all interior interfaces.**

6. **Configure redirection on the primary dirty-side switch.**

```
>> #/cfg/slb/on
>> #/cfg/slb/real 1/rip 10.1.3.2/ena
>> #/cfg/slb/real 2/rip 10.1.3.3/ena
>> #/cfg/slb/real 3/rip 10.1.3.12/ena
>> #/cfg/slb/real 4/rip 10.1.3.13/ena

>> #/cfg/slb/group 1/add 1/add 2/add 3/add 4/metric hash

>> #/cfg/slb/filt 10/dip 10.1.1.0/dmask 255.255.255.0/ena
>> #/cfg/slb/filt 20/dip 224.0.0.0/dmask 255.255.255.0/ena
>> #/cfg/slb/filt 1024/action redir/group 1/ena

>> #/cfg/slb/port a1/filt ena/add 10/add 20/ add 1024

>> #apply
>> #save
```

7. **Configure redirection on the primary clean-side switch.**

NOTE – Because the sync operation causes a panic to occur, steps 6 and 7 will need to be repeated on the secondary switches.

```
>> #/cfg/slb/on
>> #/cfg/slb/real 1/rip 10.1.2.2/ena
>> #/cfg/slb/real 2/rip 10.1.2.3/ena
>> #/cfg/slb/real 3/rip 10.1.2.12/ena
>> #/cfg/slb/real 4/rip 10.1.2.13/ena

>> #/cfg/slb/group 1/add 1/add 2/add 3/add 4/metric hash

>> #/cfg/slb/filt 10/dip 10.1.4.0/dmask 255.255.255.0/ena
>> #/cfg/slb/filt 20/dip 224.0.0.0/dmask 255.255.255.0/ena
>> #/cfg/slb/filt 1024/action redir/group 1/ena

>> #/cfg/slb/port A10/filt ena/add 10/add 20/ add 1024
```

8. **Set up SLB on the clean-side switches (optional).**

```
>> #/cfg/slb/real 5/rip 10.1.4.19/ena
>> #/cfg/slb/real 6/rip 10.1.4.20/ena
>> #/cfg/slb/group 2/add 5/add 6

>> #/cfg/slb/virt 1/ena/vip 10.1.4.100
>> #service 80/group 2

>> #/cfg/slb/port A10/server e
>> #/cfg/slb/port A1/client e
>> #/cfg/slb/port A2/client e
>> #apply
```

9. **Configure VRRP on the clean-side switches.**

   Primary Switch

```
>> #/cfg/vrrp/on
>> #/cfg/vrrp/vr 1/ena/addr 10.1.4.10/share dis/prio 101
>> #track/ifs

>> #/cfg/vrrp/vr 2/ena/vrid 2/addr 10.1.4.100/share dis/prio 101
>> #track/ifs
>> #/cfg/vrrp/vr 3/ena/vrid 3/addr 10.1.3.10/if 2
>> #apply
>> #save
```

   Secondary Switch:

```
>> #/cfg/vrrp/on
>> #/cfg/slb/sync/peer 1/ena/addr 10.1.3.1
```

10. **Configure VRRP on the dirty-side switches.**

    Primary Switch:

```
>> #/cfg/vrrp/on
>> #/cfg/vrrp/vr 1/ena/prio 101/addr 10.1.1.10/share dis
>> #track/l4pts

>> #/cfg/vrrp/vr 2/ena/vrid 2/addr 10.1.2.10

>> #/cfg/slb/sync/prios d
>> #/cfg/slb/sync/peer 1/ena/addr 10.1.1.11
>> #apply
>> #save
>> #/oper/slb/sync
```

    Secondary Switch:

```
>> #/cfg/vrrp/on
>> #/cfg/slb/sync/peer 1/ena/addr 10.1.1.1
```

11. **Verify the FWLB “sandwich” is operational.**

Four Switches, Two Routing Firewalls, with No Interior Hubs

In this configuration, all switches are active.

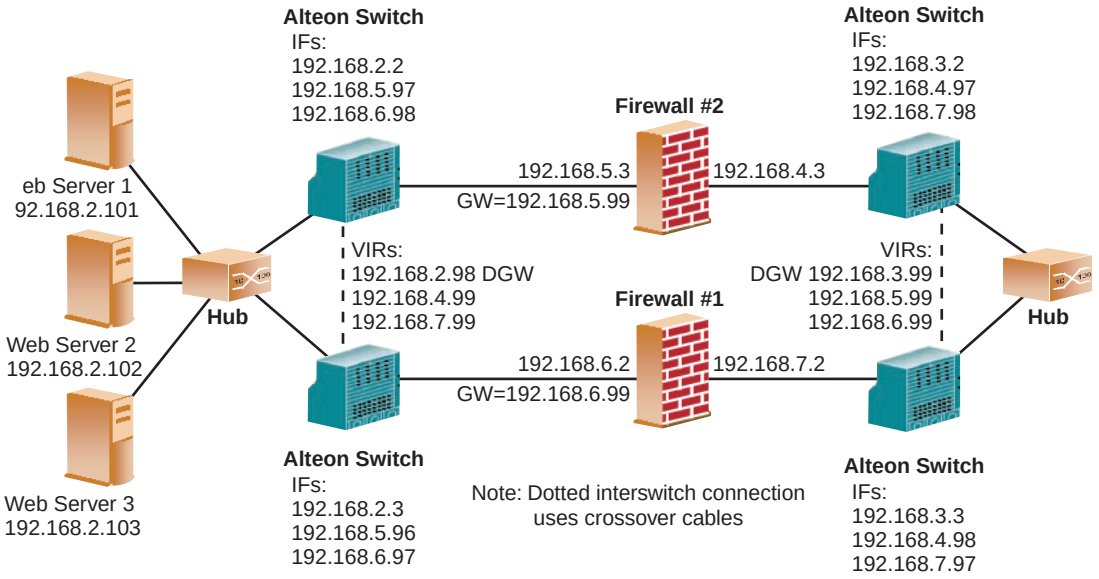


Figure 13-5 Four Switches, Two Routing Firewalls, with No Interior Hubs

---

NOTE – Throughout the following configuration dump, the sequence of commands is annotated in **bold** to indicate what is being configured at each point in the command line interface.

The configuration is broken into the following tasks:

- Client-side primary switch configuration on page 254
- Client-side secondary switch configuration on page 256
- Server-side primary switch configuration on page 256
- Server-side secondary switch configuration on page 259

Client-Side Primary Switch Configuration

```
/*  Port A5 is the ingress port for the client data.
/*  (The use of port A5 is purely arbitrary. Any port can be used.)
/*
/cfg/port A5 /ena/tag d/pvid 2/pref gig/back fast
/cfg/port A5 /fast/speed any /fctl both/mode any /auto on
/cfg/port A5 /gig/fctl both/auto on
/*
/*  VLAN 2 is created to avoid Spanning Tree issues.
/*
/cfg/vlan 2/ena/name "VLAN 2"
/cfg/vlan 2/def 5
/cfg/vlan 1/ena/name "Default VLAN"
/cfg/vlan 1/def 1 2 3 4 6 7 8 9
/*
/*  Interfaces.
/*
/cfg/ip/if 1/addr 192.168.3.2/mask 255.255.255.0/bcast 192.168.3.255/
vlan 2/ena
/cfg/ip/if 2/addr 192.168.5.97/mask 255.255.255.0/bcast 192.168.5.255/
vlan 1/ena
/cfg/ip/if 3/addr 192.168.6.98/mask 255.255.255.0/bcast 192.168.6.255/
vlan 1/ena
/*
/*  Routes to opposing-side switch for data flow.
/*
/cfg/ip/route/add 192.168.4.0 255.255.255.0 192.168.5.3
/cfg/ip/route/add 192.168.7.0 255.255.255.0 192.168.6.2
/*
/*  Turn IP forwarding on.
/*
/cfg/ip/frwd/on
/*
/*  VRRP addresses are used for health checking. This is for successful failover if a switch fails.
/*
/cfg/vrrp/on
/cfg/vrrp/track/ifs 10
/cfg/vrrp/vr 1/vrid 1/ena
/cfg/vrrp/vr 1/addr 192.168.3.99/ifs enabled
/cfg/vrrp/vr 3/vrid 3/if 2/ena
/cfg/vrrp/vr 3/addr 192.168.5.99/ifs enabled
/cfg/vrrp/vr 4/vrid 5/if 3/ena
/cfg/vrrp/vr 4/addr 192.168.6.99/ifs enabled
/*
/*  Spanning Tree is turned off. If left on, it will “see” a loop.
/*    Using a VLAN eliminates this problem.
/*
```

```
/cfg/stp/off
/*
/*  Create real servers and real server groups and set-up FWLB health checks
/*  The inter/retry/restr values used below enable fast failover, that is, one-second intervals between
/*  health checks/pings, two failed pings to remove the firewall, and four successful pings to restore
/*  the firewall to the real server group.
/cfg/slb/real 1/rip 192.168.4.99/inter 1/retry 2/restr 4/ ena
/cfg/slb/real 2/rip 192.168.7.99/inter 1/retry 2/restr 4/ena
/cfg/slb/group 1/metric hash/health icmp
/cfg/slb/group 1/add 1
/cfg/slb/group 1/add 2
/*
/*  Turn SLB on.
/*
/cfg/slb/on
/*
/*  Filter 50 allows multicast VRRP communications.
/*
/cfg/slb/filt 50/sip any/smask 0.0.0.0
/cfg/slb/filt 50/dip 224.0.0.0/dmask 255.255.255.0
/cfg/slb/filt 50/proto vrrp
/cfg/slb/filt 50/action allow/ena
/*
/*  Allow local subnet.
/*
/cfg/slb/filt 100/sip any/smask 0.0.0.0
/cfg/slb/filt 100/dip 192.168.3.0/dmask 255.255.255.0
/cfg/slb/filt 100/proto any
/cfg/slb/filt 100/action allow/ena
/*
/*  Redirect everything else through firewalls to opposing switch.
/*
/cfg/slb/filt 1024/sip any/smask 0.0.0.0
/cfg/slb/filt 1024/dip any/dmask 0.0.0.0
/cfg/slb/filt 1024/proto any
/cfg/slb/filt 1024/action redir/ena
/cfg/slb/filt 1024/group 1/rport 0
/*
/*  Add the filters to port A5, the ingress port.
/*
/cfg/slb/port A5/filt enabled
/cfg/slb/port A5/add 50
/cfg/slb/port A5/add 60
/cfg/slb/port A5/add 70
/cfg/slb/port A5/add 100
/cfg/slb/port A5/add 1024
/*
```

Client-Side Secondary Switch Configuration

The client-side secondary switch is configured with the same commands used for the primary client switch configuration. The only differences in configuration are the interfaces, as shown below:

```
/* Interfaces.
/*
/cfg/ip/if 1/addr 192.168.3.2/mask 255.255.255.0/bcast 192.168.3.255/
vlan 2/ena
/cfg/ip/if 2/addr 192.168.5.97/mask 255.255.255.0/bcast 192.168.5.255/
vlan 1/ena
/cfg/ip/if 3/addr 192.168.6.98/mask 255.255.255.0/bcast 192.168.6.255/
vlan 1/ena
/*
```

Server-Side Primary Switch Configuration

```
/* Port A5 is the ingress port for the server data.
/* (The use of port A5 is purely arbitrary. Any port can be used.)
/*
/cfg/port A5 /ena/tag d/pvid 2/pref gig/back fast
/cfg/port A5 /fast/speed any /fctl both/mode any /auto on
/cfg/port A5 /gig/fctl both/auto on
/*
/* VLAN 2 is created to avoid Spanning Tree issues.
/*
/cfg/vlan 2/ena/name "VLAN 2"
/cfg/vlan 2/def 5
/cfg/vlan 1/ena/name "Default VLAN”
/cfg/vlan 1/def 1 2 3 4 6 7 8 9
/*
/* Interfaces.
/*
/cfg/ip/if 1/addr 192.168.2.2/mask 255.255.255.0/bcast 192.168.2.255/
vlan 2/ena
/cfg/ip/if 2/addr 192.168.4.97/mask 255.255.255.0/bcast 192.168.4.255/
vlan 1/ena
/cfg/ip/if 3/addr 192.168.7.98/mask 255.255.255.0/bcast 192.168.7.255/
vlan 1/ena
/*
/* Routes to opposing side switch for data flow.
/*
/cfg/ip/route/add 192.168.5.0 255.255.255.0 192.168.4.3
/cfg/ip/route/add 192.168.6.0 255.255.255.0 192.168.7.2
/*
```

```
/*  Routes to opposing side switch for data flow.
/*
/cfg/ip/route/add 192.168.5.0 255.255.255.0 192.168.4.3
/cfg/ip/route/add 192.168.6.0 255.255.255.0 192.168.7.2
/*
/*  Turn IP forwarding on.
/*
/cfg/ip/frwd/on
/cfg/vrrp/on
/cfg/vrrp/track/ifs 10
/cfg/vrrp/vr 1/vrid 1/ena
/cfg/vrrp/vr 1/addr 192.168.2.99/ifs enabled
/*
/*  VRRP VSR for VIP failover
/*
/cfg/vrrp/vr 2/vrid 2/ena
/cfg/vrrp/vr 2/addr 192.168.2.200/ ifs enabled
/cfg/vrrp/vr 3/vrid 3/if 2/ena
/cfg/vrrp/vr 3/addr 192.168.4.99/ifs enabled
/cfg/vrrp/vr 4/vrid 5/if 3/ena
/cfg/vrrp/vr 4/addr 192.168.7.99/ifs enabled
/*
/*  Spanning Tree is turned off. If left on, it will "see" a loop.
/*  Using a VLAN eliminates this problem.
/*
/cfg/stp/off
/*
/*  Create real servers and real server groups and set-up FWLB health checks
/cfg/slb/real 1/rip 192.168.5.99/inter 1/retry 2/restr 4/ ena
/cfg/slb/real 2/rip 192.168.6.99/inter 1/retry 2/restr 4/ena
/cfg/slb/real 3/rip 192.168.2.101/inter 1/retry 2/restr 4/ena
/cfg/slb/real 4/rip 192.168.2.102/inter 1/retry 2/restr 4/ena
/cfg/slb/real 5/rip 192.168.2.103/inter 1/retry 2/restr 4/ena
/cfg/slb/group 1/metric hash/health icmp
/cfg/slb/group 1/add 1
/cfg/slb/group 1/add 2
/cfg/slb/group 2/metric roundrobin/health http
/cfg/slb/group 2/add 3
/cfg/slb/group 2/add 4
/cfg/slb/group 2/add 5
/*
/*  Virtual IP address for the Web servers.
/*
/cfg/slb/virt 1/vip 192.168.2.200 /ena
/cfg/slb/virt 1/service/http 2
/cfg/slb/virt 1/service/https/frag ena
/cfg/slb/port 1/client ena
/cfg/slb/port 5/server ena
```

```
/*
/*  Turn SLB on.
/*
/cfg/slb/on
/*
/*  Filter 50 allows multicast VRRP communications.
/*
/cfg/slb/filt 50/sip any/smask 0.0.0.0
/cfg/slb/filt 50/dip 224.0.0.0/dmask 255.255.255.0
/cfg/slb/filt 50/proto vrrp
/cfg/slb/filt 50/action allow/ena
/*
/*  Allow local subnet.
/*
/cfg/slb/filt 100/sip any/smask 0.0.0.0
/cfg/slb/filt 100/dip 192.168.2.0/dmask 255.255.255.0
/cfg/slb/filt 100/proto any
/cfg/slb/filt 100/action allow/ena
/*
/*  Redirect everything else through the firewalls to the opposing switch.
/*
/cfg/slb/filt 1024/sip any/smask 0.0.0.0
/cfg/slb/filt 1024/dip any/dmask 0.0.0.0
/cfg/slb/filt 1024/proto any
/cfg/slb/filt 1024/action redir/ena
/cfg/slb/filt 1024/group 1/rport 0
/*
/*  Add the filters for port A5, the ingress port.
/*
/cfg/slb/port A5/filt enabled
/cfg/slb/port A5/add 50
/cfg/slb/port A5/add 100
/cfg/slb/port A5/add 1024
/*
```

Server-Side Secondary Switch Configuration

The server-side secondary switch is configured with the same commands used for the primary server switch configuration. The only differences in configuration are the interfaces, as shown below:

```
/*  Interfaces.
/*
/cfg/ip/if 1/addr 192.168.2.2/mask 255.255.255.0/bcast 192.168.2.255/
vlan 2/ena
/cfg/ip/if 2/addr 192.168.4.97/mask 255.255.255.0/bcast 192.168.4.255/
vlan 1/ena
/cfg/ip/if 3/addr 192.168.7.98/mask 255.255.255.0/bcast 192.168.7.255/
vlan 1/ena

/*  All other parts are the same.
```

Firewall Configuration Example: Four Switches, Two Routing Firewalls

Using this example, shown in Figure 13-5 on page 253, the following firewall components are configured:

Firewall 1: 192.168.5.3—Internet side; 192.168.4.3—server side
Default Gateway: 192.168.5.99
Route Added: 192.168.2.0 via 192.168.4.99
Firewall 2: 192.168.6.2—Internet side; 192.168.7.2—server side
Default Gateway: 192.168.6.99
Route Added: 192.168.2.0 via 192.168.7.99

Configuring NAT on Solaris Firewalls

Network Address Translation (NAT) was set up on the firewalls facing towards the dirty side of the set-up (205.178.17.0).

To configure NAT on Checkpoint firewalls, perform the following steps:

1. **Go to "Manage Objects."**
2. **Select/create the “Network” object.**
3. **From the NAT tab, select NAT.**
4. **Click on “Add rule automatically.”**
5. **Select Hide.**

   This will hide a whole network behind a single IP address by using the range of TCP ports associated with the IP address.

6. **Enter the IP address you want to hide the network behind.**
7. **Publish the MAC address for this IP address.**

NOTE – This enables the devices requesting the IP address on ingress back from the dirty side to get the IP address. The Solaris/NT command for publishing the MAC associated with a NAT’ed IP address is:

```
arp -s <NAT IP address> <MAC address of the dirty side NIC on firewall> pub
```

Example:

```
arp -s 205.178.17.100 00:50:fe:41:fe:c0 pub
```

8. **For NAT to work with FWLB, you must assign a different NAT IP address for each firewall.**

   The default route for all firewalls was set to use 205.178.17.1. This works if you set an interface on your firewall that is on the 205.178.17.0 network. If not, it will point towards the VRRP address of the health check and the Alteon Web switch will then route to the 205.178.17.0 network.

NAT for Single Devices on the Clean-Side Network

The following procedure describes how to NAT a single server or virtual server from the clean side of the firewall to the dirty side. To do this, you need to create another filter on the dirty side that does a redirect for the NAT'ed address towards the clean side. This filter should be placed before the local subnet allow and the final redirection filters.

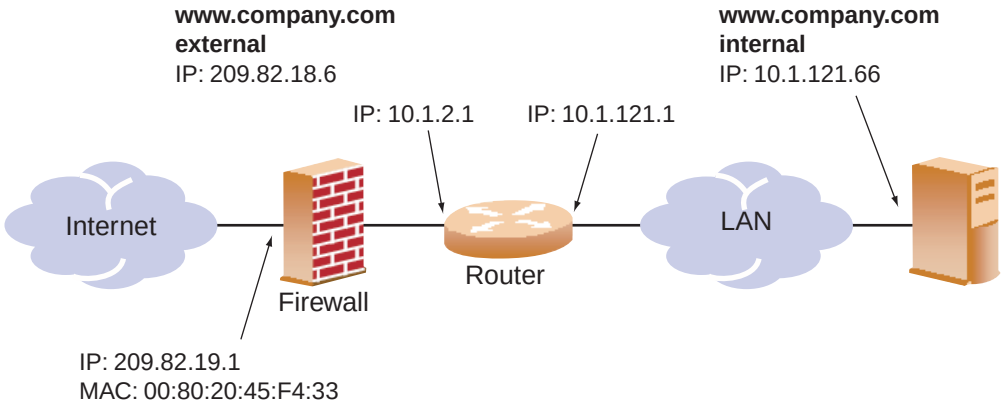


Figure 13-6 NAT for Single Devices on Clean-Side Network

You need to make the Web server at IP address 10.1.121.66 appear as 209.82.18.6 on the other side of the firewall. To do this, follow the steps below:

NOTE – All routing to the external and internal subnets should already be in place.

---

1. **Get the IP address of the system that you want to NAT (internal IP address).**

   From the diagram in Figure 13-6, this IP address is 10.1.121.66.

2. **Get the IP address of the system as it should be seen (external IP address).**

   From the diagram in Figure 13-6, this IP address is 209.82.18.6

3. **Add a routing statement.**

   You need to tell the firewall which way to send packets meant for the Web server. Even though the firewall NATs all traffic, you still need to tell the system that the external IP address is to be routed through the internal router. Do this by adding the following line to the system boot scripts:

NOTE – If your firewall is multi-netted (two or more IP addresses per NIC), do not add the specific routing statement.

```
route add host 209.82.18.6 10.1.2.1 1
```

This tells the system that IP address 209.82.18.6 is to be routed though the router located at IP address 10.1.2.1. To have this command take effect right away, you can also type it directly on the command line.

Now, if you enter the following,

```
netstat -nr | grep 209.82.18.6
```

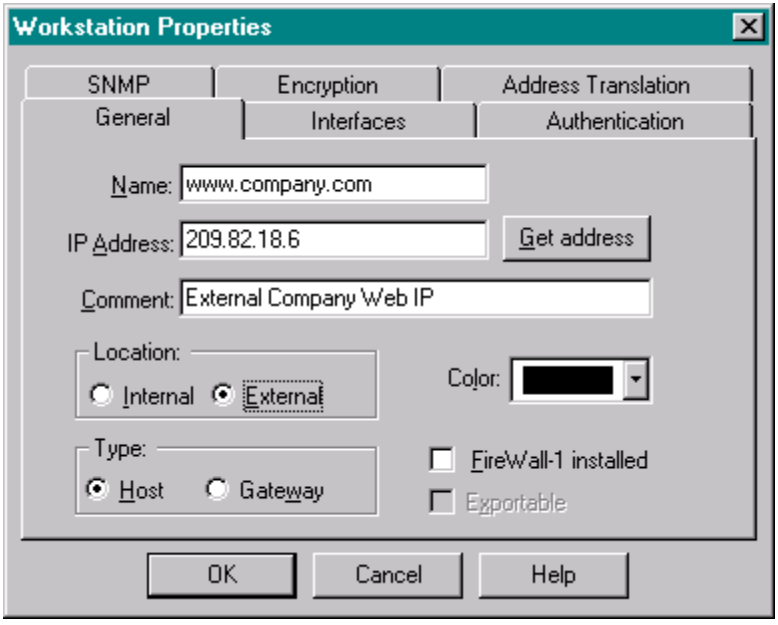
you should see the following line, indicating that the route has been successfully added.

```
209.82.18.6 10.1.2.1 UGH 0 0
```

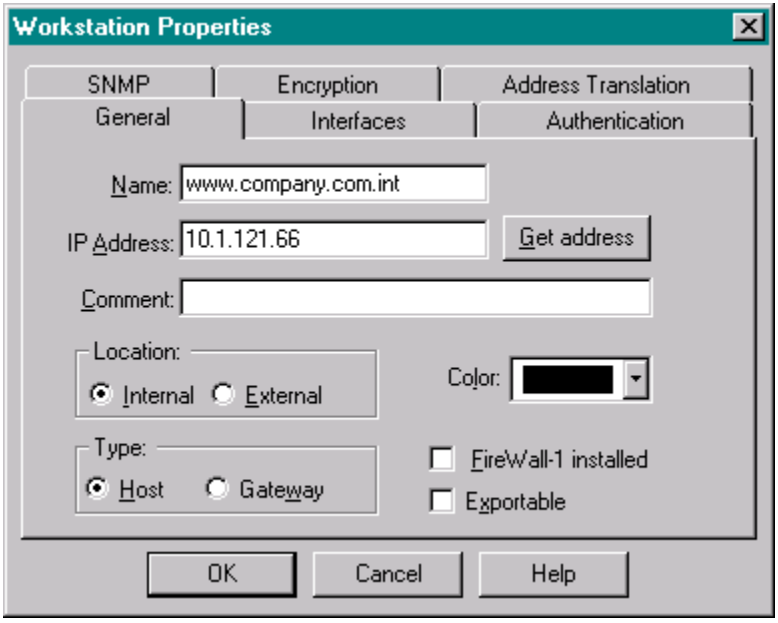
4. **Add Firewall 1 objects and NAT rules.**

Now that the system is ready, you need to tell Firewall 1 what to do. Begin by creating two new "workstation" objects.

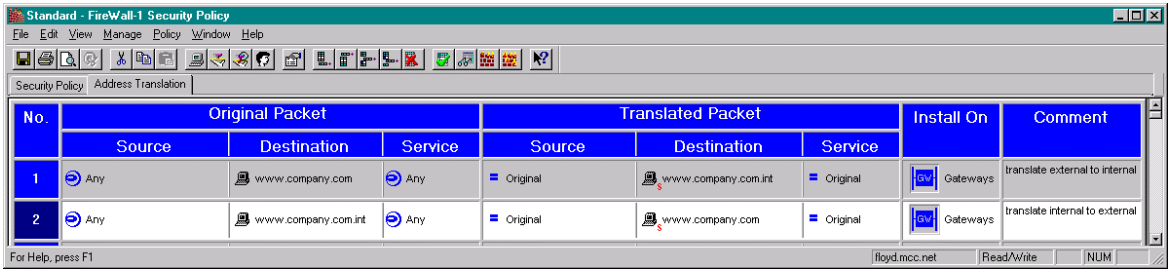
The first object is the *external* IP address ("www.company.com"), that is, how the company is known to the outside world. Enter the information as shown below:



The next object is the *internal* IP address (“www.company.com.int”), that is, how the inside network knows the company. Enter the information as shown below:



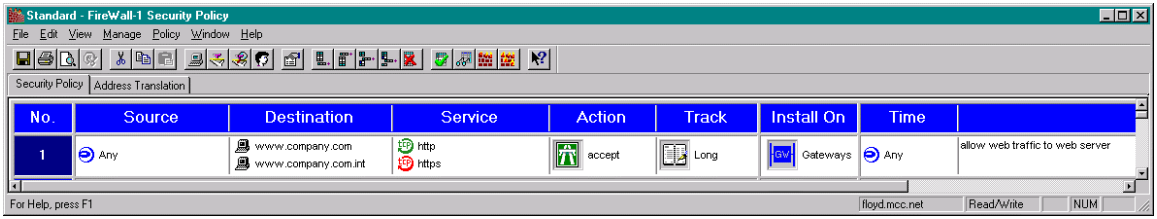
Now that two objects are created, set up Firewall 1 to do the actual NAT. Go to the Address Translation tab and enter the following:



This sets up FW #1 to do the actual translation.

NOTE – In the actual rulebase where you want to deal with these new objects, you will probably have to enter *both* objects each time you want to put either of them into a rule. Without doing this, the traffic gets rejected.

The rules that use these new objects should looks like this:



5. **Save and recompile the firewall rules.**

6. **Create a Virtual Interface Router (VIR) for the NAT'ed address.**

   This replaces the need to publish an ARP on each firewall (static NAT only).

7. **Create a NAT'ing rule on the other (clean-side) switch.**

   Create a source NAT filter on the clean-side switch. This filter will change the IP address of the data coming through the firewalls and ensure that any data coming back from the statically NAT'ed device will return to the correct firewall, be un-Nat'ed, and sent back to the dirty-side requester.

```
>> #/cfg/slb/port a8/client ena/server dis/pip 192.168.2.10
>> #/cfg/slb/filt 85/sip any/smask 255.255.255.255
>> #/cfg/slb/filt 85/dip 10.1.121.66/dmask 255.255.255.255
>> #/cfg/slb/filt 85/proto tcp/sport any/dport any
>> #/cfg/slb/filt 85/log disabled/action nat/ena
>> #/cfg/slb/filt 85/nat source
```

NOTE – You could make this a NAT rule on your CheckPoint firewall.

Static NAT, with NAT Performed by Firewall

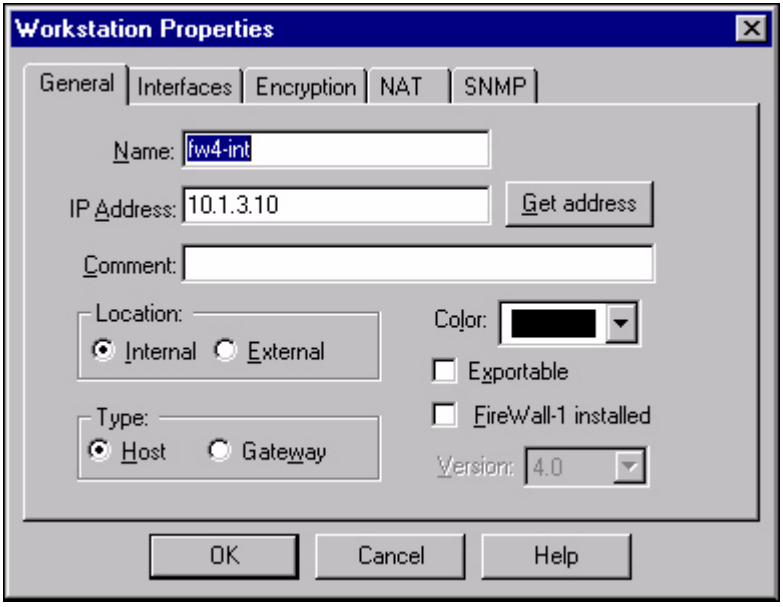
This section describes how to perform source NAT and destination NAT at the same time to data coming in from the Internet (dirty side of network) to a virtual server/virtual server router on the inside (clean) side of the network.

1. **Get the IP address of the system that you want to NAT (internal IP address).**

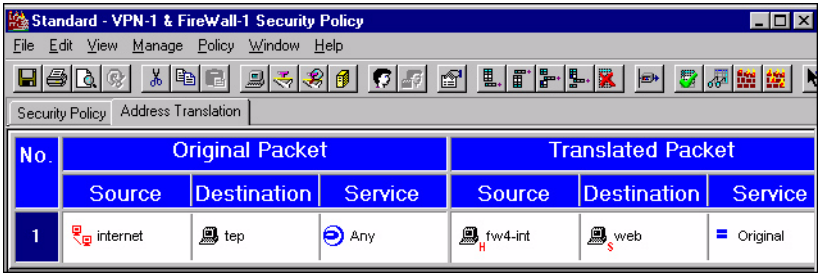
2. **Get the IP address of the system as it should be seen (external IP address).**

3. **Add Firewall 1 objects and NAT rules.**

Create workstation objects for the virtual server/virtual server router (TEP) and the interior interface of the firewall (fw4-int).



Now that we have created the two objects, we need to set up a rule that creates a range of addresses that are outside of the clean-side network (in this example, 0.0.0.1 ->255.255.255.254 (Internet) is used). Go to the Address Translation tab and enter the following.



4. **Add a routing statement.**

   You need to add a host route for the statically NAT'ed IP address. (In this example, the IP address = 205.178.17.111.) Add the following line to the system boot scripts:

   NOTE – To have this command take effect right away, you can also type it directly on the command line.

   ```
   route add host 205.178.17.111 10.1.3.1
   ```

   where IP 10.1.3.1 is the switch IP address that faces the target Web server at IP 205.178.13.123.

   The destination NAT changes anything that is destined for IP 205.178.17.111 to IP 205.178.13.123.

5. **Save and recompile the firewall rules.**

Index

Symbols

[ ] 13

Numerics

80 (port) 213
802.1Q VLAN tagging 30, 31

A

active-active redundancy 177
    configuration 187
    Server Load Balancing 189
    synchronization 199
active-standby redundancy 176
    configuration 185
administrator account 97
aggregating routes 81
    example 88
allow (filtering) 130, 133
application health checking 156
application ports 131
application redirection 145
    client IP address authentication 154
    example with NAT 150
    games and real-time applications 154
    proxies 147, 150–153
    rport 150, 153
    topologies 148
    web cache redirection example 146–154
authoritative name servers 209
Autonomous Systems 76
autonomous systems (AS) 48, 76

B

backup servers 121
BBI 48
BGP
    configuration examples 85
    failover example 85
    route aggregation 81
    selecting route paths 84
BGP attributes 83
Border Gateway Protocol (BGP) 75
    use with GSLB 229
broadcast domains 23, 29, 31, 34
Browser-Based Interface 48

C

CGI-bin scripts 110, 121
Cisco EtherChannel 41
client traffic processing 111
    SLB web balancing example 115
Command Line Interface 48
commands
    conventions used in this manual 13
configuration
    imask 117
contacting Alteon WebSystems 14
customer support 14

D

default gateway 20
  configuration example 22, 213
default password 97
deny (filtering) 130, 133
dip (destination IP address for filtering) 132
direct real server access 126
Distributed Site State Protocol (DSSP) 208, 213
dmask
  destination mask for filtering 132
domain name 217
domain name server 209
Domain Name System (DNS)
  filtering 135, 138
  Global SLB (diagram) 209
  round robin 108
dport (filtering option) 137, 151
DSSP. *See* Distributed Site State Protocol.
dynamic NAT 141

E

eBGP 76
EtherChannel 38
  as used with port trunking 41
External BGP 76
external routing 47, 76

F

failed server protection, SLB 108
failover
  active-active 182
  active-standby 174
  overview 175
fault tolerance
  port trunking 39
  Server Load Balancing 113
filtering
  allow 133
  configuration example 136
  default filter 133, 136
  deny 133
  inserting 134
  NAT configuration example 141–143
  numbering 134
  order of precedence 132, 133
  proto (option) 137, 151
  security example 135
filters
  IP address ranges 132
firewalls 135
frame tagging. *See* VLANs tagging.
FTP 222

G

gateway. *See* default gateway.
Global SLB
  configuration tutorial 213–221
  Distributed Site State Protocol 208, 213
  DNS resolution (diagram) 209
  domain name configuration 217
  health check interval 216
  hostname configuration 217
  HTTP redirect 210
  port states 215
  proxy IP address 223
  real server groups 214
  real servers 214
  remote site configuration 216
  tests 225
  using proxy IP 224
GSLB
  using proxy IP 222

H

hash metric 119
health checks 150
  configuration using scripts 164
  format 162
  Global SLB interval 216
  IMAP server parameters 161
  RADIUS server parameters 160
  real server parameters 156
  verifying scripts 166
high availability 169
host routes
  configuration example 68
hostname, for HTTP health checks 158, 217
hot-standby redundancy 177
  configuration 197
HTTP
  application health checks 158
  redirects (Global SLB option) 210

I

iBGP 76
ICMP 130
IEEE standards
    802.1Q VLAN tagging 30, 31
IF. *See* IP interfaces.
IGMP 130
IMAP server health checks 161
incoming route maps 79
inserting filters 134
internal BGP 76
internal routing 47, 76
Internet Service Provider (ISP), SLB configuration example 112
inter-switch port states 179
IP address
    conservation 141
    filter ranges 132
    local route cache ranges 25
    private 141
    proxies 110, 123, 147, 150–153
    real server groups 114, 191, 214
    real servers 109, 113, 214
    routing example 21
    virtual servers 109, 110, 115, 215
IP interfaces
    configuration example 213
    example configuration 21, 24, 114
    routing 18
    VLAN #1 (default) 31
    VLANs 31
IP proxies
    for application redirection 153
    for Server Load Balancing 123
    See also proxies, proxy IP address (PIP).
IP routing 111
    cross-subnet example 18
    default gateway configuration 22
    IP interface configuration 21, 24
    IP interfaces 18
    IP subnets 19
    network diagram 19
    subnet configuration example 21
    switch-based topology 20
IP subnets 18, 20
    routing 18, 19, 20
    VLANs 29, 31
ISL Trunking 38

L

Layer 4
    administrator account 97
    optional software 107
least connections (SLB Real Server metric) 120
lmask (local route cache parameter) 25
lnet (local route cache parameter) 25
local route cache
    IP address ranges for 25
local route cache parameters
    lmask 25
    lnet 25
log
    filtering option 130, 135
logical segment. *See* IP subnets.

M

Management Processor (MP) 31
    use in switch security 92
manual style conventions 13
mapping virtual ports to real ports 122
maxcon limit 121
maximum connections 121
minimum misses (SLB real server metric) 118
MP (Management Processor) 31
multi homing 229
multi links between switches
    using port trunking 38
    using VLANs 34

N

name servers, Global SLB configuration example 209
NAT. *See* Network Address Translation.
Network Address Translation (NAT) 130, 150
    configuration example 141–143
    filter example 142
    proxy 142
    static example 143
network performance
    statistics, with use of proxy addresses 124
NFS server 112
noncacheable sites
    application redirection 154
none (port processing mode)
    SLB web balancing example 115
non-HTTP redirects
    proxy IP address 223

O

optional software 145
  Layer 4 SLB support 107
OSPF 130
  configuration examples 59
  creating a virtual link 62
  defining an OSPF domain 60
  defining host routes 68
  redistributing routes 78, 82
  route maps 78, 79
  summarizing routes 66
  verifying configuration 73
outgoing route maps 79
overflow servers 121

P

parallel links 34
password
  administrator account 97
  default 97
  L4 administrator account 97
  user account 97
PIP. *See* proxies, proxy IP address.
POP3 222
port 80 213
port processing mode
  client 115
  none 115
  server 111, 115
port states 215
port trunking 38
  configuration example 40
  description 41
  EtherChannel 38
  fault tolerance 39
ports
  for services 131
  mapping 127
  physical. *See* switch ports.
private IP address 141
private network 141
protocol types 130
proxies 110, 123, 147–153
  configuration example 142
  NAT 141
proxy IP address
  configuring in GSLB 224
  in Global SLB 223
proxy IP address (PIP) 110, 123, 126, 153
proxy servers 147
PVID (port VLAN ID) 30

R

RADIUS server parameters 160
real server groups
  backup/overflow servers 121
  configuration example 114, 191, 214
real servers 110
  backup/overflow servers 121
  configuration example 214
  connection timeouts 120
  health checks 156
  maximum connections 121
  weights 120
redirect (HTTP) 210
redirecting non-HTTP applications 222
redirection. *See* application redirection
redistributing routes 78, 82, 88
redundancy
  active-active 177
  active-standby 176
  hot-standby 177
remote (Global SLB real server property) 216
roundrobin
  SLB Real Server metric 120
route aggregation 81, 88
route maps 78
  configuring 79
  incoming and outgoing 79
route paths in BGP 84
routers 19, 22
  border 48, 76
  peer 48, 76
  port trunking 38
  switch-based routing topology 20
  using redirection to reduce Internet congestion 145, 146
  web cache redirection example 146
routes, advertising 48, 76
routing
  internal and external 47, 76
rport
  filtering 150, 153

S

scalability, service 108
security
  filtering 130, 135
  firewalls 135
  private networks 141
  switch management 91
  VLANs 29
segmentation. *See* IP subnets.
segments. *See* IP subnets.
server (port processing mode)
  SLB web balancing example 115
Server Load Balancing
  across subnets 18
  active-active redundancy 189
  backup servers 121
  complex network topologies 123
  configuration example 112, 123
  direct real server access 126
  distributed sites 208
  failed server protection 108
  fault tolerance 113
  health checks 156
  maximum connections 121
  overflow servers 121
  overview 109
  port processing modes 111
  proxies 110, 123
  proxy IP addresses 126
  real server group 114, 191
  real server IP address (RIP) 109
  real servers 110
  remote sites 208
  topology considerations 110
  virtual IP address (VIP) 109, 110
  virtual servers 109
  weights 120
server pool 108
server port processing 111
server traffic processing
  Global SLB configuration example 215
service ports 131
shared services 108
SIP (source IP address for filtering) 132
smask
  source mask for filtering 132
SMTP 222
SNMP 48
Spanning Tree Protocol
  eliminating bridge loops with VRRP 201
  replacing with hot-standby failover 197
  VLANs 34, 35
spoofing, prevention of 91
sport (filtering option) 137, 151
static NAT 143
statistical load distribution 38
summarizing routes
  example 66
switch failover 175
switch management
  security 91
  via IP interface 31
switch ports
  VLANs membership 30
syslog
  messages 130

T

tagging. *See* VLANs tagging.
TCP 130, 138, 139
  health checking using 118
  port 80 127
TCP/UDP
  port numbers 122
Telnet 135
text conventions 13
timeouts
  real server connnections 120
transparent proxies 123, 147, 150–153
typographic conventions, manual 13

U

UDP 130, 138, 139
  server status using 118
user account 97
Using proxy IP
  GSLB 222

V

virtual interface router (VIR) 170
virtual IP address (VIP) 109, 110
virtual link
  configuration example 62
Virtual Local Area Networks. *See* VLANs.
virtual router
  ID numbering 202
virtual router group 178
virtual server router 181
virtual servers 109
  IP address 115, 215
VLAN tagging
  See Also VLANs tagging.
VLANs 148
  broadcast domains 23, 29, 31, 34
  default 30
  example showing multiple VLANs 32
  ID numbers 30
  IP interface configuration 24
  IP interfaces 31
  Management Processor 31
  multiple links 34
  multiple VLANs 30, 31
  parallel links example 34
  port configuration 23
  port members 30
  PVID 30
  routing 23
  security 29
  Spanning Tree Protocol 34, 35
  supporting multiple VLANs 32
  tagging 30–33
  topologies 31
  VLAN #1 (default) 30–31, 149, 213, 218
  VLAN-tagged adapter support 31
VRRP (Virtual Router Redundancy Protocol)
  active-active redundancy 177
  active-standby redundancy 176
  hot-standby redundancy 177
  inter-switch port states 179
  overview 170
  synchronization 199
  synchronizing configurations 179
  tracking 183
  virtual interface router 170
  virtual router ID numbering 202
  virtual server router 181
  vrid 170

W

web cache redirection. *See* application redirection
web cache servers 146–149
web hosting 112
weights 120
World Wide Web, client security for browsing 135