

---

# Foundry ServerIron® Switch Installation and Configuration Guide



**FOUNDRY**  
**NETWORKS**

2100 Gold Street  
P.O. Box 649100  
San Jose, CA 95164-9100  
Tel 408.586.1700  
Fax 408.586.1900

June 2002

---

---

Copyright © 2002 Foundry Networks, Inc. All rights reserved.

No part of this work may be reproduced in any form or by any means – graphic, electronic or mechanical, including photocopying, recording, taping or storage in an information retrieval system – without prior written permission of the copyright owner.

The trademarks, logos and service marks ("Marks") displayed herein are the property of Foundry or other third parties. You are not permitted to use these Marks without the prior written consent of Foundry or such appropriate third party.

Foundry Networks, BigIron, FastIron, IronView, *JetCore*, NetIron, ServerIron, *TurboIron*, *IronWare*, *EdgeIron*, the Iron family of marks and the Foundry Logo are trademarks or registered trademarks of Foundry Networks, Inc. in the United States and other countries.

F-Secure is a trademark of F-Secure Corporation. All other trademarks mentioned in this document are the property of their respective owners.

---

## CHAPTER 1

### **GETTING STARTED..... 1-1**

INTRODUCTION .....	1-1
AUDIENCE .....	1-1
NOMENCLATURE .....	1-1
RELATED PUBLICATIONS .....	1-2
WHAT'S NEW IN THIS EDITION? .....	1-2
SERVER LOAD BALANCING (SLB) ENHANCEMENTS .....	1-2
GLOBAL SERVER LOAD BALANCING (GSLB) ENHANCEMENTS .....	1-6
TRANSPARENT CACHE SWITCHING ENHANCEMENTS .....	1-7
URL SWITCHING ENHANCEMENTS .....	1-7
SYSTEM-LEVEL ENHANCEMENTS .....	1-7
HOW TO GET HELP .....	1-9
WEB ACCESS .....	1-9
EMAIL ACCESS .....	1-9
TELEPHONE ACCESS .....	1-9
WARRANTY COVERAGE .....	1-9

## CHAPTER 2

### **FEATURES OVERVIEW ..... 2-1**

HARDWARE FEATURES .....	2-1
CHASSIS-BASED SERVERIRONS .....	2-1
STACKABLE SERVERIRONS .....	2-3
LEDs .....	2-3
SOFTWARE FEATURES .....	2-4
FLASH IMAGE .....	2-4
DETERMINING THE FLASH VERSION THE SERVERIRON IS RUNNING .....	2-4
SERVERIRON FEATURE LIST .....	2-5
GLOBAL NETWORK ADDRESS TRANSLATION .....	2-7
ACCESS AND MANAGEMENT FEATURES .....	2-8

MANAGEMENT INTERFACES .....	2-8
MULTIPLE LEVELS OF ACCESS CONTROL .....	2-10
DYNAMIC CONFIGURATION .....	2-11
SOFT REBOOT .....	2-11
SCHEDULED SYSTEM RELOAD .....	2-11
TELNET .....	2-11
TRIVIAL FILE TRANSFER PROTOCOL (TFTP) .....	2-12
SIMPLE NETWORK TIME PROTOCOL (SNTP) .....	2-12
DOMAIN NAME SERVER (DNS) RESOLVER .....	2-12
SNMPV2C SUPPORT .....	2-12
REMOTE MONITORING (RMON) STATISTICS .....	2-13
SYSLOG LOGGING .....	2-13
PING AND TRACE-ROUTE FACILITIES .....	2-13
PORT MIRRORING .....	2-14
BASIC LAYER 3 SERVICES .....	2-14
PROXY SERVER CACHE LOAD BALANCING .....	2-14
SELECTABLE QUALITY OF SERVICE (QOS) .....	2-14
SLB MULTINETTING USING NETWORK ADDRESS TRANSLATION (NAT) .....	2-15
IP FILTERS .....	2-16
TCS USES OF FILTERS .....	2-17
PORT PROFILES .....	2-18
HEALTH CHECKING .....	2-19
ICMP MESSAGE FEATURE FOR HTTP .....	2-19
FASTCACHE .....	2-19
SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP) TRAPS .....	2-20
GRACEFUL OR FORCED SERVER SHUTDOWN .....	2-21
HOT STANDBY REDUNDANCY .....	2-21
SERVER LOAD BALANCING (SLB) FEATURES .....	2-21
VALUE OF SLB .....	2-22
CONFIGURABLE LOAD-BALANCING PREDICTOR .....	2-23
CONFIGURABLE APPLICATION GROUPING .....	2-23
UNLIMITED VIRTUAL IP ADDRESSES (VIPs) .....	2-25
GEOGRAPHICALLY-DISTRIBUTED SERVERS .....	2-25
GLOBAL SERVER LOAD BALANCING (GSLB) .....	2-26
SYMMETRIC SERVER LOAD BALANCING .....	2-26
SWITCHBACK .....	2-27
MANY-TO-ONE TCP/UDP PORT BINDING .....	2-29
HTTP REDIRECT .....	2-29
TRANSPARENT VIP AND STATELESS APPLICATION PORTS .....	2-29
WEB SWITCHING FEATURES .....	2-29
URL SWITCHING .....	2-30
COOKIE SWITCHING .....	2-31
HTTP HEADER HASHING .....	2-32
SSL SESSION ID SWITCHING .....	2-33
TRANSPARENT CACHE SWITCHING (TCS) FEATURES .....	2-34
HOW TCS WORKS .....	2-34

---

ADVANCED STATISTICS .....	2-36
CACHE ROUTE OPTIMIZATION (CRO) .....	2-36
POLICY-BASED CACHE FAILOVER (CFO) .....	2-36
FIREWALL LOAD BALANCING .....	2-37
IRONCLAD FIREWALL LOAD BALANCING .....	2-38
IP FORWARDING .....	2-38
NETWORK ADDRESS TRANSLATION (NAT) .....	2-38
LAYER 2 SWITCHING FEATURES .....	2-38
MAC SWITCHING .....	2-38
STATIC MAC ENTRIES .....	2-39
STANDARD SPANNING TREE PROTOCOL (STP) .....	2-39
IRONSPAN STP ENHANCEMENTS .....	2-40
TRUNK GROUPS .....	2-40
PORT-BASED VIRTUAL LANS (VLANs) .....	2-40
VLAN TAGGING .....	2-40
MAC FILTERS .....	2-40
ADDRESS-LOCK FILTERS .....	2-41
DYNAMIC HOST CONFIGURATION PROTOCOL (DHCP) ASSIST .....	2-41
IP MULTICAST CONTAINMENT .....	2-41

## CHAPTER 3

<b>INSTALLING THE SERVERIRON .....</b>	<b>3-1</b>
UNPACKING A SYSTEM .....	3-1
PACKAGE CONTENTS .....	3-1
GENERAL REQUIREMENTS .....	3-1
SUMMARY OF INSTALLATION PROCEDURES .....	3-1
INSTALLATION PRECAUTIONS .....	3-2
PREPARING THE INSTALLATION SITE .....	3-3
CABLING INFRASTRUCTURE .....	3-3
INSTALLATION LOCATION .....	3-3
VERIFYING PROPER OPERATION .....	3-3
ATTACHING A PC OR TERMINAL .....	3-3
ASSIGNING IP ADDRESSES .....	3-5
MOUNTING THE SERVERIRON IN AN EQUIPMENT RACK .....	3-6
DESKTOP INSTALLATION .....	3-6
RACK MOUNT INSTALLATION .....	3-6
POWERING ON THE SERVERIRON .....	3-7
CONNECTING NETWORK DEVICES .....	3-7
CABLE LENGTH .....	3-8
CONNECTING TO ETHERNET OR FAST ETHERNET HUBS .....	3-9
CONNECTING TO WORKSTATIONS, SERVERS, OR ROUTERS .....	3-9
TROUBLESHOOTING NETWORK CONNECTIONS .....	3-9
TESTING CONNECTIVITY .....	3-10
PINGING AN IP ADDRESS .....	3-10
TRACING A ROUTE .....	3-10

CONFIGURING A SYSTEM NAME, CONTACT, AND LOCATION .....	3-10
MANAGING THE SERVERIRON .....	3-11
LOGGING ON THROUGH THE CLI .....	3-11
LOGGING ON THROUGH THE WEB MANAGEMENT INTERFACE .....	3-11
ESTABLISHING SNMP COMMUNITY STRINGS .....	3-11
ENCRYPTION OF SNMP COMMUNITY STRINGS .....	3-12
ADDING AN SNMP COMMUNITY STRING .....	3-12
DISPLAYING THE SNMP COMMUNITY STRINGS .....	3-12

## CHAPTER 4

### USING THE WEB SWITCHING MANAGEMENT MODULE..... 4-1

WSM CPU LOAD SHARING .....	4-2
EXAMPLE CONFIGURATION 1 .....	4-3
EXAMPLE CONFIGURATION 2 .....	4-4
DISPLAYING THE SLOT ALLOCATIONS FOR THE WSM CPUS .....	4-5
CHANGING SLOT ALLOCATIONS .....	4-5
ENABLING THE WSM CPUS .....	4-5
BROADCASTING SESSION DELETE MESSAGES TO WSM CPUS .....	4-6
UPGRADING THE SOFTWARE .....	4-6
UPGRADING THE MP BOOT CODE .....	4-6
UPGRADING THE WSM CPU BOOT CODE .....	4-6
UPGRADING THE MP FLASH CODE .....	4-7
UPGRADING THE WSM CPU FLASH CODE .....	4-7
CHANGING THE DEFAULT BOOT SOURCE .....	4-8
CHANGING THE MANAGEMENT SESSION FROM THE MP TO A WSM CPU .....	4-9
LOGGING IN TO A WSM CPU .....	4-9
ENTERING COMMANDS TO THE WSM CPU .....	4-10
LOGGING OUT FROM THE WSM CPU .....	4-10
WSM CPU COMMANDS .....	4-10
TEMPERATURE SENSOR .....	4-11
DISPLAYING THE TEMPERATURE .....	4-11
DISPLAYING TEMPERATURE MESSAGES .....	4-12
CHANGING TEMPERATURE WARNING AND SHUTDOWN LEVELS .....	4-13
CHANGING THE CHASSIS POLLING INTERVAL .....	4-14
CONFIGURING REDUNDANCY .....	4-15
SWITCHOVER .....	4-15
MANAGEMENT SESSIONS .....	4-16
SYSLOG AND SNMP TRAPS .....	4-16
MAC ADDRESS CHANGES .....	4-16
CONFIGURING THE REDUNDANT MANAGEMENT PARAMETERS .....	4-16
INSERTING THE MODULE .....	4-18
DETERMINING REDUNDANT MANAGEMENT MODULE STATUS .....	4-18
DISPLAYING SWITCHOVER MESSAGES .....	4-19
FILE SYNCHRONIZATION BETWEEN THE ACTIVE AND STANDBY REDUNDANT MANAGEMENT MODULES ...	4-19
SWITCHING OVER TO THE STANDBY REDUNDANT MANAGEMENT MODULE .....	4-23

---

DISPLAYING WEB SWITCHING MANAGEMENT MODULE INFORMATION .....	4-24
DISPLAYING THE SOFTWARE VERSION RUNNING ON THE MODULE .....	4-24
DISPLAYING GENERAL MODULE INFORMATION .....	4-25
DETERMINING MODULE STATUS .....	4-26
DETERMINING THE SLOT ALLOCATIONS FOR THE WSM CPUS .....	4-29
CHANGING SLOT ALLOCATIONS FOR THE WSM CPUS .....	4-30
ADDITIONAL DISPLAY COMMANDS .....	4-30

## CHAPTER 5

### CONFIGURING HOT STANDBY REDUNDANCY ..... 5-1

HOW HOT STANDBY REDUNDANCY WORKS .....	5-1
FAILOVER IN SOFTWARE RELEASE 07.3.03 AND LATER .....	5-2
CONFIGURING TWO SERVERIRONs FOR HOT STANDBY .....	5-2
CONFIGURING A BACKUP GROUP ID .....	5-4
CHANGING THE BACKUP TIMER .....	5-5
VIEWING HOT STANDBY INFORMATION .....	5-6
HOT STANDBY CONFIGURATION EXAMPLE .....	5-7
CONFIGURING THE BACKUP PORT FOR SLB REDUNDANCY .....	5-11
DISPLAYING HOT STANDBY CONFIGURATION INFORMATION .....	5-12

## CHAPTER 6

### CONFIGURING SERVER LOAD BALANCING ..... 6-1

SLB PARAMETERS .....	6-2
CONFIGURATION GUIDELINES .....	6-11
BASIC CONFIGURATION EXAMPLE .....	6-11
DEFINING THE REAL SERVERS AND ADDING THE APPLICATION PORTS .....	6-12
DEFINING THE VIRTUAL SERVER .....	6-17
BINDING VIRTUAL AND REAL SERVERS .....	6-20
CONFIGURING GLOBAL SLB PARAMETERS .....	6-21
FAST-PATH SLB PROCESSING .....	6-22
LOAD BALANCING METHOD (PREDICTOR) .....	6-24
ROUTER PORTS .....	6-27
TCP SYN LIMIT .....	6-27
WARNING AND SHUTDOWN THRESHOLDS .....	6-28
ICMP UNREACHABLE MESSAGES .....	6-29
SENDING A TCP RST OR ICMP UNREACHABLE MESSAGE TO A CLIENT .....	6-30
SOURCE IP ADDRESS .....	6-30
SOURCE NAT .....	6-32
REVERSE NAT .....	6-33
FORCE SHUTDOWN .....	6-34
STICKY AGE .....	6-34
PERSISTENT STICKY CONNECTIONS .....	6-35
TRANSPARENT VIP .....	6-36
TCP FAST AGING .....	6-36
DECREMENTING THE CURRENT CONNECTION COUNTER IMMEDIATELY FOLLOWING A SERVER RST .....	6-36

CONFIGURING REAL SERVER PARAMETERS .....	6-37
IP ADDRESS .....	6-37
LOCATION .....	6-37
BACKUP SERVER .....	6-38
APPLICATION PORTS .....	6-40
HOST RANGES AND HOST-RANGE MAPS .....	6-41
MAXIMUM CONNECTIONS .....	6-45
TRAFFIC RATE THRESHOLD .....	6-46
WARNING AND SHUTDOWN THRESHOLDS .....	6-46
LAYER 3 HEALTH CHECK .....	6-47
SOURCE NAT .....	6-47
WEIGHT .....	6-48
CONFIGURING REAL SERVER APPLICATION PORT PARAMETERS .....	6-49
PORT STATE .....	6-50
BINDING STATE .....	6-50
KEEPALIVE HEALTH CHECK STATE .....	6-50
CONNECTION RATE LIMITING .....	6-51
LAYER 7 HEALTH CHECK PARAMETERS .....	6-52
CONFIGURING VIRTUAL SERVER PARAMETERS .....	6-52
APPLICATION PORTS AND BINDINGS .....	6-52
PRIMARY AND BACKUP SERVERS .....	6-53
HOST RANGE .....	6-54
HTTP REDIRECT .....	6-54
LOAD BALANCING METHOD (PREDICTOR) .....	6-55
SYMMETRIC SLB PRIORITY .....	6-55
TRACK PORTS .....	6-56
TRACK PORT GROUP .....	6-57
ENABLING SERVER CLUSTER SUPPORT .....	6-59
FAST AGING FOR UDP SESSIONS .....	6-59
NORMAL UDP AGING FOR DNS AND RADIUS .....	6-60
TRANSPARENT VIP .....	6-60
CONFIGURING VIRTUAL SERVER APPLICATION PORT PARAMETERS .....	6-60
PORT STATE .....	6-60
STICKY .....	6-61
CONCURRENT .....	6-61
SWITCHBACK (DSR) .....	6-62
SMOOTH FACTOR .....	6-62
STATELESS .....	6-64
VIRTUAL SOURCE .....	6-64
TRANSLATION .....	6-65
ENHANCED SSL ACCELERATOR SUPPORT .....	6-66
CONFIGURING AN IP FILTER FOR A TCP/UDP PORT .....	6-67
SHUTTING DOWN A REAL SERVER .....	6-69
VIEWING SLB CONFIGURATION DETAILS AND STATISTICS .....	6-69
DISPLAYING GLOBAL CONFIGURATION INFORMATION .....	6-70
DISPLAYING REAL SERVER INFORMATION .....	6-74



---

DISPLAYING VIRTUAL SERVER INFORMATION .....	6-82
DISPLAYING PORT-BINDING INFORMATION .....	6-92
DISPLAYING SESSION STATISTICS .....	6-93
DISPLAYING TRAFFIC STATISTICS .....	6-94
SLB APPLICATION EXAMPLES .....	6-96
WEB HOSTING WITH ONE VIRTUAL SERVER MAPPED TO MULTIPLE REAL SERVERS .....	6-97
WEB HOSTING WITH MULTIPLE VIRTUAL SERVERS MAPPED TO ONE REAL SERVER .....	6-97
MANY-TO-ONE TCP/UDP PORT BINDING .....	6-98
WEB HOSTING WITH UNLIMITED VIRTUAL IP ADDRESSES .....	6-101
SLB INTRANET CONFIGURATION WITH HTTP, TELNET HOSTING ACROSS MULTIPLE VIRTUAL SERVERS AND MULTIPLE REAL SERVERS .....	6-104
TCP/UDP APPLICATION GROUPS .....	6-104
WEB HOSTING WITH SERVERIRON AND REAL SERVERS IN DIFFERENT SUB-NETS .....	6-107
WEB HOSTING WITH GEOGRAPHICALLY-DISTRIBUTED SERVERS .....	6-110
USING HTTP REDIRECT WITH GEOGRAPHICALLY-DISTRIBUTED SERVERS .....	6-113
LOAD BALANCING STREAMING MEDIA FILES .....	6-119
GLOBALLY DISABLING TCP OR UDP PORTS .....	6-121

## CHAPTER 7

### CONFIGURING SYMMETRIC SLB

#### AND SWITCHBACK ..... 7-1

USING SYMMETRIC SERVER LOAD BALANCING .....	7-1
ACTIVE-STANDBY SSLB .....	7-1
ACTIVE-ACTIVE SSLB .....	7-8
CHANGING THE SSLB DISCOVERY INTERVAL .....	7-10
USING DYNAMIC SSLB PRIORITY .....	7-10
DISPLAYING SYMMETRIC SLB INFORMATION .....	7-14
DELAYING AN SSLB SERVERIRON'S ACTIVATION FOLLOWING RECOVERY .....	7-15
USING SWITCHBACK .....	7-15
USING REMOTE FAILOVER SERVERS FOR SWITCHBACK .....	7-16
USING HEALTH CHECKS WITH SWITCHBACK .....	7-16
SWITCHBACK CONFIGURATION EXAMPLE .....	7-17
CLI COMMANDS .....	7-18
CONFIGURING THE LOOPBACK ADDRESS ON A REAL SERVER .....	7-20

## CHAPTER 8

### CONFIGURING TRANSPARENT VIPs

#### AND STATELESS SLB ..... 8-1

TRANSPARENT VIP .....	8-1
STATEFUL AND STATELESS LOAD BALANCING .....	8-4
ENABLING THE TRANSPARENT VIP FEATURE .....	8-4
CONFIGURING AN INDIVIDUAL VIRTUAL SERVER TO BE TRANSPARENT .....	8-4
COMPLETE CLI EXAMPLE FOR FIGURE 8.1 ON PAGE 8-2 .....	8-4
COMPLETE CLI EXAMPLE FOR FIGURE 8.2 ON PAGE 8-3 .....	8-6

STATELESS TCP/UDP PORTS .....	8-8
HOW THE SERVERIRON SELECTS A REAL SERVER FOR A STATELESS PORT .....	8-9
CONFIGURING A STATELESS APPLICATION PORT .....	8-10
STATELESS HEALTH CHECKING .....	8-10
CONFIGURING STATELESS HEALTH CHECKS .....	8-12

## CHAPTER 9

### CONFIGURING GLOBAL SERVER LOAD BALANCING..... 9-1

GSLB OVERVIEW .....	9-2
THE GSLB POLICY .....	9-5
CONFIGURING GSLB .....	9-9
CONFIGURING THE PROXY .....	9-11
ENABLING THE GSLB PROTOCOL ON THE SITE SERVERIRONS .....	9-18
SPECIFYING THE GSLB SITES AND THE SITE SERVERIRONS .....	9-19
SPECIFYING THE DNS ZONES AND THE HOST APPLICATIONS .....	9-21
MODIFYING GSLB PROTOCOL PARAMETERS .....	9-26
CHANGING THE GSLB PROTOCOL PORT NUMBER .....	9-27
CHANGING THE GSLB PROTOCOL UPDATE PERIOD .....	9-28
MODIFYING THE GSLB PARAMETERS RELATED TO DNS RESPONSES .....	9-29
MODIFYING THE GSLB POLICY PARAMETERS .....	9-33
CONFIGURING AFFINITY .....	9-44
CONFIGURING DNS CACHE PROXY .....	9-47
ENABLING DNS CACHE PROXY .....	9-48
DISPLAYING THE DNS CACHE PROXY STATE .....	9-48
DISPLAYING STATISTICS FOR TRANSPARENT DNS QUERY INTERCEPT OR DNS CACHE PROXY .....	9-49
COMBINING THE DNS CACHE PROXY AND DNS OVERRIDE FEATURES .....	9-49
CONFIGURING TRANSPARENT DNS QUERY INTERCEPT .....	9-50
DISPLAYING GSLB INFORMATION .....	9-55
DISPLAYING SITE INFORMATION .....	9-56
DISPLAYING SERVER INFORMATION FOR A REMOTE SERVERIRON .....	9-61
DISPLAYING ZONE AND HOST NAME INFORMATION .....	9-62
DISPLAYING DETAILED DNS ZONE INFORMATION .....	9-65
DISPLAYING THE GSLB POLICY .....	9-67
DISPLAYING RTT PREFIX CACHE ENTRIES .....	9-71
DISPLAYING GSLB RESOURCE UTILIZATION .....	9-73
DISPLAYING DYNAMIC CONFIGURATION INFORMATION .....	9-75
SNMP TRAPS AND SYSLOG MESSAGES .....	9-77
DISPLAYING THE SYSLOG MESSAGES .....	9-79
DISABLING AND RE-ENABLING TRAPS .....	9-79

## CHAPTER 10

### CONFIGURING TRANSPARENT CACHE SWITCHING ..... 10-1

CONFIGURING TCS .....	10-1
TCS PARAMETERS .....	10-2
CONFIGURATION NOTES .....	10-3

---

EXAMPLE TCS APPLICATION .....	10-3
ENABLING TCS .....	10-4
ADDING WEB CACHE SERVERS .....	10-7
ASSIGNING WEB CACHE SERVERS TO A CACHE GROUP .....	10-8
DISABLING A CACHE GROUP OR A SERVER WITHIN A CACHE GROUP .....	10-10
REMOVING OR RE-ASSIGNING AN INTERFACE .....	10-11
DEFINING DISTRIBUTION OF WEB REQUESTS WITHIN A CACHE GROUP .....	10-12
ADDING A VIRTUAL IP ADDRESS FOR POLICY-BASED CACHE FAILOVER (CFO) .....	10-14
CACHE SERVER SPOOFING .....	10-14
MODIFYING GLOBAL TCS PARAMETERS .....	10-16
MODIFYING DEFAULT SETTINGS FOR CACHE SERVERS .....	10-17
MODIFYING DEFAULT TCP/UDP PORT PARAMETERS .....	10-23
USING IP FILTERS TO CONTROL CACHING .....	10-24
DISPLAYING CONFIGURATION INFORMATION AND STATISTICS .....	10-27
SHUTTING DOWN A CACHE SERVER .....	10-31
TCS APPLICATION EXAMPLES .....	10-32
BASIC TCS CONFIGURATION EXAMPLE .....	10-32
POP BELONGING TO AN ISP USING CACHING TO MINIMIZE WAN COSTS .....	10-35
CACHE ROUTE OPTIMIZATION (CRO) .....	10-35
POLICY-BASED CACHING .....	10-37
FASTCACHE .....	10-39
POLICY-BASED CACHE FAILOVER (CFO) .....	10-41
PROXY SERVER CACHE LOAD BALANCING .....	10-42
CONTENT AWARE CACHE SWITCHING .....	10-45
STREAMING MEDIA SUPPORT .....	10-60

## CHAPTER 11

### CONFIGURING LAYER 7 SWITCHING ..... 11-1

CONFIGURING URL SWITCHING .....	11-1
BASIC URL SWITCHING EXAMPLE .....	11-2
URL SWITCHING EXAMPLE FOR TWO WEB SITES USING ONE VIP .....	11-13
USING URL SWITCHING TO DIRECT HTTP REQUESTS TO SPECIFIC TCP PORTS .....	11-18
CONFIGURING COOKIE SWITCHING .....	11-22
SETTING UP THE SERVERS .....	11-24
CONFIGURING THE SERVER TO SET A COOKIE .....	11-24
ENABLING COOKIE SWITCHING ON THE VIRTUAL SERVER .....	11-25
USING URL SWITCHING AND COOKIE SWITCHING CONCURRENTLY .....	11-27
SETTING UP URL SWITCHING POLICIES .....	11-28
CONFIGURING SERVER GROUPS AND SERVER IDS .....	11-28
CONFIGURING THE SERVER TO SET A COOKIE .....	11-29
ENABLING CONCURRENT URL AND COOKIE SWITCHING ON THE VIRTUAL SERVER .....	11-29
CONFIGURING HTTP HEADER HASHING .....	11-30
COOKIE HASHING .....	11-30
SELECTIVE COOKIE HASHING .....	11-32
URL STRING HASHING .....	11-35

URL SEGMENT HASHING .....	11-37
DISPLAYING HASHING BUCKET STATISTICS .....	11-40
CONFIGURING SSL SESSION ID SWITCHING .....	11-40
CONFIGURING REAL SERVERS FOR SSL .....	11-42
CONFIGURING THE VIRTUAL SERVER FOR SSL SESSION ID SWITCHING .....	11-43
SETTING THE SSL AGING PERIOD .....	11-46
SETTING THE MAXIMUM NUMBER OF DATABASE ENTRIES .....	11-47
VIEWING LAYER 7 SWITCHING DETAILS AND STATISTICS .....	11-47
DISPLAYING URL SWITCHING POLICY INFORMATION .....	11-47
DISPLAYING LAYER 7 SWITCHING STATISTICS .....	11-49
SETTING THE MAXIMUM NUMBER OF LAYER 7 SWITCHING CONNECTIONS .....	11-52
USING HTTP 1.1 FOR CONNECTIONS TO REAL SERVERS .....	11-52
DROPPING HTTP REQUESTS WHEN A SERVER GROUP REACHES MAX CONNECTIONS .....	11-53
DROPPING HTTP REQUESTS WHEN A SERVER GROUP IS UNAVAILABLE .....	11-53

## CHAPTER 12

### CONFIGURING PORT AND

### HEALTH CHECK PARAMETERS ..... 12-1

CONFIGURING HEALTH CHECKS .....	12-1
LAYER 3 HEALTH CHECKS .....	12-2
LAYER 4 HEALTH CHECKS .....	12-3
LAYER 7 HEALTH CHECKS .....	12-3
HEALTH CHECKING FOR REAL SERVERS IN OTHER SUB-NETS .....	12-3
HEALTH CHECK SUMMARY .....	12-4
SERVER AND APPLICATION PORT STATES .....	12-15
REASSIGN THRESHOLD .....	12-19
CONFIGURING HEALTH CHECK PARAMETERS .....	12-19
CUSTOMIZING LAYER 7 HEALTH CHECKS .....	12-31
CHECKING THE HEALTH OF MULTIPLE WEB SITES ON THE SAME REAL SERVER .....	12-39
USING A LAYER 7 HEALTH CHECK FOR AN UNKNOWN PORT .....	12-40
CONFIGURING BOOLEAN HEALTH-CHECK POLICIES (SERVERIRON 400 AND SERVERIRON 800) .....	12-41
CONFIGURING BOOLEAN HEALTH-CHECK POLICIES (SERVERIRONXL) .....	12-52
VIEWING APPLICATION PORT STATUS IN THE SYSLOG .....	12-57
CONFIGURING SESSION TABLE PARAMETERS .....	12-57
MODIFYING MAXIMUM SESSION LIMIT .....	12-58
MODIFYING THE TCP AGE .....	12-59
MODIFYING THE UDP AGE .....	12-59
MODIFYING THE CLOCK SCALE .....	12-60
ENABLING SYSLOG MESSAGES FOR SESSION TABLE ENTRIES .....	12-60
CONFIGURING THE SLOW-START MECHANISM .....	12-62
OVERVIEW .....	12-62
PORT SLOW-START MECHANISM .....	12-64
DISABLING THE SLOW START MECHANISM .....	12-68

---

## CHAPTER 13

### CONFIGURING IP FORWARDING ..... 13-1

PRODUCT SUPPORT .....	13-1
OVERVIEW .....	13-1
EXAMPLES .....	13-2
CONFIGURING IP FORWARDING .....	13-4
SUPPORTED CONFIGURATIONS .....	13-7
CONFIGURING IP INTERFACES AND ROUTE TABLE ENTRIES .....	13-7
ADDING A VIRTUAL ROUTING INTERFACE .....	13-8
ADDING AN IP INTERFACE TO THE VIRTUAL ROUTING INTERFACE .....	13-8
ADDING A STATIC IP ROUTE .....	13-9
ADDING A STATIC ARP ENTRY .....	13-10
ENABLING IP FORWARDING .....	13-10
CONFIGURING RIP .....	13-11
CONVERTING AN EXISTING CONFIGURATION TO WORK WITH IP FORWARDING .....	13-13
DISPLAYING IP FORWARDING CONFIGURATION INFORMATION AND STATISTICS .....	13-13
DISPLAYING THE IP FORWARDING STATE .....	13-14
DISPLAYING ARP ENTRIES .....	13-15
DISPLAYING IP INTERFACES .....	13-17
DISPLAYING THE IP ROUTE TABLE .....	13-18
DISPLAYING IP TRAFFIC STATISTICS .....	13-19
IP FORWARDING APPLICATION EXAMPLES .....	13-20
USING THE SERVERIRON AS A DEFAULT GATEWAY .....	13-20
FORWARDING TRAFFIC FROM ONE SUB-NET TO ANOTHER .....	13-22
FORWARDING MANAGEMENT ACCESS FROM A DIFFERENT SUB-NET .....	13-23
FORWARDING TRAFFIC BETWEEN A CLIENT AND VIP ON DIFFERENT SUB-NETS .....	13-24
FORWARDING SYMMETRIC SLB TRAFFIC BETWEEN SUB-NETS .....	13-26
FORWARDING SWITCHBACK TRAFFIC BETWEEN SUB-NETS .....	13-28
FORWARDING IN AN SLB HOT-STANDBY CONFIGURATION .....	13-30
FORWARDING TRAFFIC BETWEEN SUB-NETS IN A TRANSPARENT CACHE SWITCHING CONFIGURATION .....	13-32
FORWARDING TRAFFIC IN A NETWORK ADDRESS TRANSLATION CONFIGURATION .....	13-33
FORWARDING TRAFFIC IN A GSLB CONFIGURATION .....	13-33
FORWARDING TRAFFIC IN A HIGH-AVAILABILITY FWLB CONFIGURATION .....	13-35

## CHAPTER 14

### CONFIGURING NETWORK ADDRESS TRANSLATION ..... 14-1

PORT ADDRESS TRANSLATION .....	14-2
PROTOCOLS SUPPORTED FOR NAT .....	14-3
CONFIGURING NAT .....	14-4
CONFIGURING STATIC ADDRESS TRANSLATIONS .....	14-4
CONFIGURING DYNAMIC NAT PARAMETERS .....	14-5
ENABLING NAT .....	14-6
CHANGING TRANSLATION TABLE TIMEOUTS .....	14-6
DISPLAYING THE ACTIVE NAT TRANSLATIONS .....	14-7
DISPLAYING NAT STATISTICS .....	14-8

CLEARING TRANSLATION TABLE ENTRIES .....	14-9
NAT DEBUG COMMANDS .....	14-10
CONFIGURING NAT USING THE WEB MANAGEMENT INTERFACE .....	14-11
CONFIGURING GLOBAL NAT PARAMETERS .....	14-11
CONFIGURING STATIC NAT ENTRIES .....	14-13
CONFIGURING DYNAMIC NAT ENTRIES .....	14-14

## **APPENDIX A**

### **USING SYSLOG .....A-1**

OVERVIEW .....	A-1
DISPLAYING SYSLOG MESSAGES .....	A-2
ADDITIONAL SYSLOG CONFIGURATION INFORMATION .....	A-2
SYSLOG MESSAGES .....	A-2

## **APPENDIX B**

### **NETWORK MONITORING .....B-1**

CONFIGURING RMON .....	B-1
STATISTICS (RMON GROUP 1) .....	B-1
HISTORY (RMON GROUP 2) .....	B-2
ALARM (RMON GROUP 3) .....	B-2
EVENT (RMON GROUP 9) .....	B-3
MONITORING LAYER 4 STATISTICS .....	B-3
LAYER 4 STATISTICS GROUP .....	B-4
LAYER 4 HISTORY GROUP .....	B-4
VIEWING SYSTEM INFORMATION .....	B-6
VIEWING CONFIGURATION INFORMATION .....	B-7
VIEWING PORT STATISTICS .....	B-7
VIEWING STP STATISTICS .....	B-7
CLEARING STATISTICS .....	B-8

## **APPENDIX C**

### **HTTP STATUS CODES .....C-1**

---

# Chapter 1

## Getting Started

### Introduction

This guide describes the Foundry Networks ServerIron® switch products. Procedures are provided for installing the hardware and configuring the software. The software procedures show how to perform tasks using the CLI and using the Web management interface.

---

**NOTE:** The ServerIron provides most of the Layer 2 features that other Foundry switches provide, in addition to Server Load Balancing (SLB), Global Server Load Balancing (GSLB), Transparent Cache Switching (TCS), and Firewall Load Balancing (FWLB).

---

### Audience

This guide is designed for system administrators with a working knowledge of Layer 2 and Layer 4 switching.

### Nomenclature

This guide uses the following typographical conventions to show information:

*Italic* highlights the title of another publication and occasionally emphasizes a word or phrase.

**Bold** highlights a CLI command.

***Bold Italic*** highlights a term that is being defined.

Underline highlights a link on the Web management interface.

Capitals highlights field names and buttons that appear in the Web management interface.

---

**NOTE:** A note emphasizes an important fact or calls your attention to a dependency.

---

---

**WARNING:** A warning calls your attention to a possible hazard that can cause injury or death.

---

---

**CAUTION:** A caution calls your attention to a possible hazard that can damage equipment.

---

## Related Publications

The following Foundry Networks documents supplement the information in this guide.

- *Foundry ServerIron Firewall Load Balancing Guide* – provides detailed feature descriptions, procedures, and application examples for Firewall Load Balancing (FWLB).
- *Foundry ServerIron Command Line Interface Reference* – provides detailed syntax information for all ServerIron CLI commands.
- *Foundry Switch and Router Installation and Basic Configuration Guide* – provides an overview of the Foundry switching and routing products and their features. The ServerIron provides many of the Layer 2 switching features of other Foundry switches, so configuration procedures for the Layer 2 features also apply to the ServerIron.

To order additional copies of the manuals, do one of the following:

- Call 1-877-TURBOCALL (887-2622) in the United States or 408.586.1881 outside the United States.
- Send email to [info@foundrynet.com](mailto:info@foundrynet.com).

## What's New in This Edition?

This edition describes the following software releases:

- 07.3.04 for the ServerIronXL (contains Layer 3 support)
- 07.2.25 for the ServerIron 400 and ServerIron 800
- 07.1.21 for the ServerIronXL

The following tables list the new features and changes since the last edition of the documentation.

**NOTE:** For information about FWLB enhancements, see the *Foundry ServerIron Firewall Load Balancing Guide*.

## Server Load Balancing (SLB) Enhancements

Enhancement	Description	Release	See
New load-balancing predictor	The <b>least-local-conn</b> predictor enables each WSM CPU on a ServerIron 400 or ServerIron 800 to make load balancing decisions based on the number of active connections on the individual WSM CPU.	07.2.25	6-24
Normal UDP aging for DNS and RADIUS	The <b>port dns   radius udp-normal-age</b> command ages DNS or RADIUS sessions using the UDP age timer. By default, DNS and RADIUS sessions are immediately deleted when a reply is received from the server.	07.2.25	6-60
Option to disable the Layer 3 health check for an individual real server	You can configure the ServerIron to make a real server's state ACTIVE without pinging the server's IP address first.	07.2.25	6-47 12-20



Enhancement	Description	Release	See
Destination Unreachable messages to clients supported for stateless SLB	<p>In previous releases, for stateful SLB you can configure the ServerIron to send an ICMP Destination Unreachable message to a client if the client's request for a TCP application on a VIP is not successful. Release 07.2.25 extends this support to stateless SLB.</p> <p>In addition, the default behavior is changed. In the current release, the ServerIron sends a TCP RST to the client if the requested port is not available. In previous releases, the ServerIron quietly dropped the request.</p>	07.2.25	6-30
Using SSL health checks in a health check policy.	You can use SSL health checks as part of a health check policy.	07.2.25	12-46
Recursive lookups for DNS health checks	You can enable the DNS real server to recursively look up the IP address or zone name in the health check. By default, the server does not perform recursive lookups to resolve a health check.	07.2.25	12-28
SLB optimization	You can enable the ServerIron to use a fast-path for processing stateful or stateless SLB.	07.2.24	6-22
Real server traffic threshold	You can set a threshold for the number of bytes per second sent and received on a real server. Once this threshold is reached, the ServerIron assigns no further connections to the real server.	07.2.24	6-46
New command to disable the reassignment counter	You can disable the reassignment counter, which is incremented when a server does not respond to a client's TCP SYN.	07.2.24	12-29
Scripted health checks	You can configure content verification health checks for ports that do not use one of the well-known port numbers recognized by the ServerIron.	07.2.24	12-36
Broadcasting session delete messages to WSM CPUs	When a session is maintained on multiple WSM CPUs, you can configure the server's WSM CPU to broadcast a session delete message to the other WSM CPUs when it deletes the server session.	07.2.24	4-6
TCP fast aging	Following a RST from the server, the ServerIron ages out session table entries in the amount of time specified in the <b>server msl</b> command, by default 8 seconds. You can optionally configure the ServerIron to use the 1 – 2 minute aging time used in previous releases.	07.2.24	6-36
Decrementing the current connection counter immediately after server RST	You can configure the ServerIron to immediately decrement its current connection counter when it receives a RST from the server.	07.2.24	6-36

Enhancement	Description	Release	See
New command to send TCP RST messages to clients	You can configure the ServerIron to send a TCP RST to a client if the client's request for a TCP application on a VIP is not successful.  <b>Note:</b> In 07.2.24, this enhancement applies only to stateful SLB. Beginning in 07.2.25, the enhancement also applies to stateless SLB.	07.2.24	6-30
Enhanced SSL Accelerator support	You can configure a ServerIronXL to return traffic from a real server back to the SSL accelerator from which it was sent rather than sending it directly to the client.	07.2.25 07.3.04	6-66
Host-range maps	In previous releases, the mapping between VIP host range addresses on a virtual server and host range addresses on real servers had to be sequential and contiguous. In release 07.3.04, addresses in the host range on the real server(s) no longer need to be contiguous.	07.3.04	6-41
New load-balancing predictor	The new <b>least-sess</b> predictor selects the server that has the fewest active session entries in the ServerIron's session table. The <b>least-conn</b> predictor used to have this meaning but now selects the server with the fewest active connections with clients.	07.3.03	6-24
SLB hot standby is more sensitive to changes in the health of server ports	Failover occurs even if the active and standby ServerIrons have the same number of healthy router ports, when the active ServerIron has fewer healthy server ports. In previous releases, failover in this situation occurs only if the active ServerIron has zero healthy server ports.	07.3.03	5-2
Primary and backup servers	You can designate each real or remote server as a primary server or backup server. SLB load balances among the primary servers, regardless of whether they are added as real servers or remote servers. SLB uses the backup servers only if all the primary servers are unavailable.	07.3.04 07.2.23	6-38 6-53
Boolean health-check policies	You can associate Layer 4 and Layer 7 health checks with individual applications on individual servers, and create policies that join the health checks using the Boolean operators AND and OR.  <b>Note:</b> Boolean health check policies also are supported in software release 07.3.02. However, the syntax and options are different from those in 07.2.23.	07.2.23	12-41
Server cluster support	You can configure the ServerIron to stop sending traffic on an established connection to a server when the requested application is down on the server. This feature is useful in server cluster configurations such as a Network File System (NFS) server farm.	07.2.23	6-59

Enhancement	Description	Release	See
New command to disable all application ports on a real server	You can use the <b>port disable-all</b> command at the real server CLI level to disable all the application ports on the server.	07.2.23	6-50
New command to unbind all application ports on a real server	You can use the <b>port unbind-all</b> command at the real server CLI level to unbind all the application ports on the server from their VIPs. Optionally, you can specify a particular VIP from which to unbind the ports.	07.2.23	6-50
New SSLB command to delay reactivation following recovery after a failover	You can configure a ServerIron in an SSLB configuration to delay reactivation after it recovers from a failover. Delaying reactivation minimizes interruption to sessions created by the standbyServerIron for the VIPs owned by the recovered ServerIron.	07.3.04 07.2.24 07.1.21	7-15
Active-active SSLB support	You can configure both ServerIrons in an SSLB configuration to actively load balance requests for the same VIP.	07.2.15	7-8
Configurable thresholds for application response time	You can configure warning and shutdown thresholds for real servers. The ServerIron generates Syslog messages and SNMP traps if an application's average response time exceeds a threshold. The ServerIron shuts down an application that exceeds the shutdown threshold.	07.2.20	6-28
Support for cloning real servers	For easy administration, you can clone existing real servers, then change individual server parameters as needed.	07.2.20	6-16
Option to change a real server's IP address	You can change the IP address of a real server without completely removing and re-adding the server configuration.	07.2.20	6-37
Option to disable the stateless SLB hashing algorithm for UDP ports	You can disable the stateless SLB hashing algorithm for UDP connections consisting of one client packet and one server response packet, and use the round-robin load balancing method to select a real server instead.	07.2.20	8-10
Persistent sticky connections	You can configure the ServerIron to accept new sessions for the same real server for a sticky port, even when the port is waiting to be unbound or deleted, or is disabled.	07.1.20	6-35
Fast aging for UDP sessions	You can configure the ServerIron to delete UDP sessions immediately after a server sends a response to a client.	07.2.20	6-59
VIP-based health check policies	You can associate Layer 3 and Layer 4 health checks with individual virtual IP addresses (VIPs).	07.3.02	12-52
Option to disable the Layer 3 health check for real servers	You can configure the ServerIron to make a real server's state ACTIVE without pinging the server's IP address first.	07.2.20 07.1.20	12-20

Enhancement	Description	Release	See
Support for basing an alias port's health on the health of its master port	You can configure an alias port to base its own health on the health of its master port. By default, the ServerIron checks the health of alias ports and their master ports independently.	07.1.20	12-26
Support for TCP Layer 7 health checks on unknown ports	You can use Layer 7 health checks for SMTP, LDAP, Telnet, POP3, IMAP4, FTP on ports other than the well-known ports for these applications. Previous releases support the health checks only on the well-known ports.	07.2.20 07.1.20	12-40
Support for Layer 7 DNS health checks on unknown ports	You can use the Layer 7 health check parameters configured for the DNS port (UDP port 53) to check the health of other, unknown, UDP ports.	07.1.18	12-41
New SSL health check for ServerIronXL	The SSL health checking procedure has changed.  <b>Note:</b> This change applies to the ServerIronXL only.  The new SSL health checking procedure for the ServerIronXL initiates an SSL connection with the server on TCP port 443, negotiates a secure link, and transfers encrypted data across it.	07.1.18	12-14
Option to use simple SSL health checks	You can configure the ServerIronXL to use the SSL health check method used in software releases previous to 07.1.18, instead of the new method implemented in 07.1.18 (see above).	07.1.20	12-30
Slow start following failed Layer 7 health check	When a real server is restored after having failed a Layer 7 health check, it is brought back up using the slow-start mechanism.	07.2.20	12-62
Option to globally disable the slow-start mechanism	You can globally disable the slow-start mechanism, without removing the slow-start configuration information.	07.1.20	12-68
Option to clear a real server's sessions to force deletion of the real server	You can force the ServerIron to clear all the sessions for a real server from the session table by entering the <b>clear server session</b> <name> command. When you delete a real server, all the server's sessions must be cleared before the ServerIron will completely remove the real server.	07.3.04 07.2.20	CLI Reference

## Global Server Load Balancing (GSLB) Enhancements

Enhancement	Description	Release	See
DNS response selection counters are shown with site information	The <b>show gslb dns zone</b> and <b>show gslb dns detail</b> commands show DNS response selection counters.	07.2.24	9-62

Enhancement	Description	Release	See
Connection limit metric	The connection limit metric selects among otherwise equal sites based on the weighted average number of new connections per second over a specified interval.	07.2.25 07.3.04	9-38

## Transparent Cache Switching Enhancements

Enhancement	Description	Release	See
Streaming media over TCS	You can configure TCS for the RTSP, MMS, and Real streaming media protocols.	07.2.20 07.1.20	10-60

## URL Switching Enhancements

Enhancement	Description	Release	See
Wildcard character support for Host header information	The <b>port http url-host-id</b> command in a URL switching configuration now supports using an asterisk (*) as a wildcard character to specify one or more characters at the beginning of the Host header field.	07.3.02	11-16

## System-Level Enhancements

Enhancement	Description	Release	See
Higher default maximum number of MP sessions	The default maximum number of session table entries on the MP is increased to 1,000,000. In previous releases, the default maximum number of session table entries is 524,288.  The maximum configurable number of session table entries for the ServerIron 400 and ServerIron 800 is still 2,000,000.	07.2.25	12-58
Syslog messages indicate the reason an application port has gone down	Syslog messages generated when a real server application port goes down indicate the reason the port is down. Previous releases list the IP address and port that went down, but not the reason the port went down.  The new format of the message is:  L4 server <ip-addr> <name> is down due to <reason>	07.2.25	A-2
Management module redundancy	You can configure a pair of management modules in the chassis as an active-standby pair.	07.2.24	4-15

Enhancement	Description	Release	See
Option to suppress Telnet connection rejection messages	You can disable the message that the Foundry device sends to a Telnet client that is denied access to the device.	07.2.24	CLI Reference
Support for SSH access control using ACLs	You can more tightly control management access to a device through SSH by using ACLs.	07.3.04	Security Guide
TCP out of buffer logging	You can configure the ServerIron to log an ALERT message when either the TCP send buffer or TCB memory is exhausted.	07.3.04	CLI Reference
VLAN option for active-active ports	You can use a tagged port for the synchronization link in an active-active configuration for SSLB, SYN-Guard, or NAT. This enhancement applies to the following commands: <ul style="list-style-type: none"> <li><b>server backup-port</b></li> <li><b>server active-active-port</b></li> </ul>	07.2.23 07.1.21	7-9 CLI Reference
Support for using ACLs to filter debug output	You can use an ACL to filter output from <b>debug</b> commands.	07.2.23	Diag Guide
Packet capture utility	This release features a utility that captures packets directed to the ServerIron's CPU, based on user-defined filters. The captured packets are stored in a capture buffer and can be displayed onscreen or transferred to a TFTP server.	07.2.23	Diag Guide
<b>show chassis</b> output added to the <b>show tech</b> command	The <b>show tech</b> command now includes output for the <b>show chassis</b> command.	07.2.23	Does not affect the manuals
SYN-Defense™	SYN-Defense enhances TCP SYN attack protection so that the ServerIron 400 or ServerIron 800 can complete the three-way handshake on behalf of a connecting client.	07.2.20	Security Guide
SYN-Guard™	Using the SYN-Guard feature, you can configure the ServerIron to terminate new connection requests for a server on the ServerIron itself, and send only the established connections to the server. This provides protection against Denial of Service (DoS) attacks.	07.2.20	Security Guide
TCP SYN protection on an individual port basis	You can enable TCP SYN protection on specific ports and leave the feature disabled on the remaining ports.	07.2.15	Security Guide
Connection Rate Limiting (CRL) for individual applications	You can specify the maximum number of new TCP or UDP connections a real server, firewall, or cache server can have. You also can specify the maximum number of new connections for individual applications.	07.2.20	6-51 10-24 FWLB Guide
Option to globally disable TCP or UDP ports	You can globally disable a Layer 4 port on the ServerIron. The port can be disabled for all real servers, all virtual servers, or all real and virtual servers.	07.2.20	6-121

Enhancement	Description	Release	See
Description option for servers, firewalls, and caches	You can add a description to a real server, virtual server, firewall, or cache. The description appears in the output of various show commands and in the running-config and startup-config files.	07.2.20	CLI Reference

## How to Get Help

Foundry Networks technical support will ensure that the fast and easy access that you have come to expect from your Foundry Networks products will be maintained.

### Web Access

- <http://www.foundrynetworks.com>

### Email Access

Technical requests can also be sent to the following email address:

- [support@foundrynet.com](mailto:support@foundrynet.com)

### Telephone Access

- 1-877-TURBOCALL (887-2622)      United States
- 408.586.1881      Outside the United States

## Warranty Coverage

Contact Foundry Networks using any of the methods listed above for information about the standard and extended warranties.





---

## Chapter 2

# Features Overview

This chapter provides an overview of the ServerIron Layer 2 and Layer 4 switching features, and Layer 3 IP forwarding features.

Foundry Networks' ServerIron switch provides Internet service providers (ISPs) and enterprise Intranet managers with a high-density, high-performance Layer 4 switch that improves the performance of existing servers, ensures applications availability and increases network redundancy.

The ServerIron can be used in all server farm or transparent caching environments, including Web, FTP and email. The ServerIron provides 10-, 100-, and 1000-Mbps connectivity to servers. You also can multi-home links by configuring trunk groups for greater bandwidth and redundancy.

---

**NOTE:** This guide focuses on the Layer 4 Server Load Balancing (SLB), Web Switching, and Transparent Cache Switching (TCS) features of the ServerIron. For Firewall Load Balancing (FWLB) information, see the *Foundry ServerIron Firewall Load Balancing Guide*. For configuration details for the Layer 2 features, see the *Foundry Switch and Router Installation and Basic Configuration Guide*.

---

## Hardware Features

The ServerIron switch is available in fixed-port stackable configurations and in multi-slot chassis configurations.

---

**NOTE:** This guide refers to all these products as "ServerIron" except where the architectural features of a specific ServerIron model are relevant.

---

### Chassis-Based ServerIrons

The ServerIron 400 and ServerIron 800 are chassis-based Layer 4 – 7 web switching devices. They provide the rich feature set of Foundry Networks's other ServerIron products, in two chassis-based multi-slot platforms.

Figure 2.1 on page 2-2 shows the ServerIron 400.

**Figure 2.1 ServerIron 400**

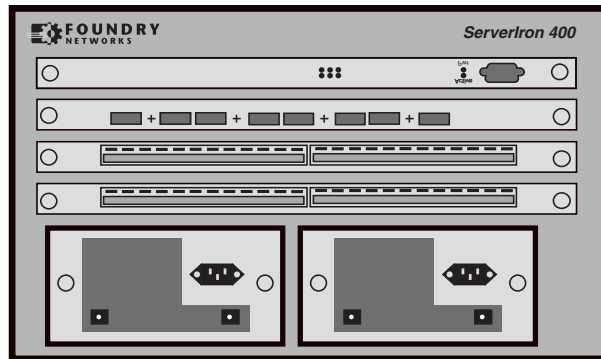
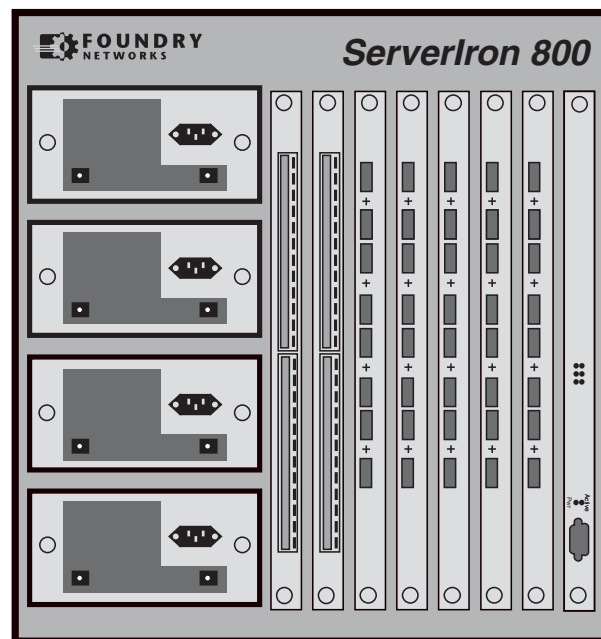


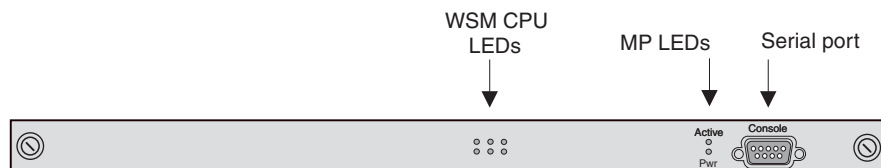
Figure 2.2 shows the ServerIron 800.

**Figure 2.2 ServerIron 800**



The ServerIron 400 and ServerIron 800 are powered by the Web Switching Management Module, a multi-processor management module. Figure 2.3 shows the Web Switching Management Module.

**Figure 2.3 Web Switching Management Module**



The Web Switching Management Module does not have network interfaces but does have a serial management interface. In addition, the module has status LEDs for the Management Processor (MP) and the web switching module CPUs (WSM CPUs).

See “Using the Web Switching Management Module” on page 4-1 for more information.

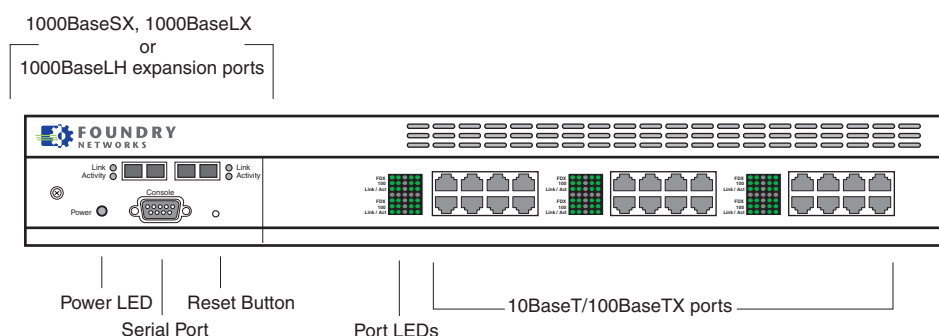
## Stackable ServerIrons

The following stackable configurations are available:

- 8-port ServerIronXL (model number FCSLB8) – provides eight 10/100 Base-Tx ports.
- 16-port ServerIronXL (model number FCSLB16) – provides 16 10/100 Base-Tx ports.
- 24-port ServerIronXL (model number FCSLB24) – provides 24 10/100 Base-Tx ports.
- 8-port ServerIronXL/G (model number T8SLBF) – provides eight 1000 Base-Sx Gigabit ports.

These models are 2-RU (rack-unit) high. For 10/100 stackables, Gigabit Ethernet expansion modules are available in 1000BaseSX multi-mode and 1000BaseLX single-mode. Figure 2.4 shows an example of the FCSBL24.

**Figure 2.4 ServerIronXL 24-port switch with a 2-port expansion module installed**



## LEDs

The ServerIron indicates port hardware status using the following LEDs.

**Table 2.1: Port LED indicators for ServerIron Gigabit ports**

LED	Position	State	Meaning
Link	Top	On	Port is connected.
		Off	No port connection exists.
Activity	Bottom	Blinking	Traffic is being transmitted or received on the port.

**Table 2.2: Port LED indicators for ServerIron 10BaseT/100BaseTX ports**

LED	Position	State	Meaning
FDX/HDX	Top	On	The port is operating at full-duplex.
		Off	The port is operating at half-duplex.

**Table 2.2: Port LED indicators for ServerIron 10BaseT/100BaseTX ports**

LED	Position	State	Meaning
100	Middle	On	The port is operating at 100 Mbps.
		Off	The port is not operating at 100 Mbps.
Link/Act	Bottom	On	The port is connected.
		Off	No port connection exists.
		Blinking	Traffic is being transmitted or received on that port.

In addition, the green Power LED is lit when the power supply is active.

For information about the LEDs on the Web Switching Management Module, see “Status LEDs” on page 4-27.

## Software Features

The following sections summarize the ServerIron software.

### Flash Image

The flash image (system software) that is running on a device determines the software features that are supported by that device. Table 2.3 lists the ServerIron image file names. In each image name, “xxxxx” represents the release number.

**Table 2.3: Foundry ServerIron Software Images**

Product	Flash image	Description
ServerIron 400 ServerIron 800	WSMxxxxx.bin (management processor code) WSPxxxxx.bin (web switching CPU code)	ServerIron 400/800 code
ServerIronXL (8-, 16-, and 24-port models)	SLBxxxxx.BIN	Stackable ServerIron code
ServerIronXL/G	BSIxxxxx.BIN	Stackable ServerIron code

### Determining the Flash Version the ServerIron Is Running

To determine the flash image running on the ServerIron, do one of the following.

#### USING THE CLI

Enter the following command: **show version**

**NOTE:** You can enter this command from any CLI access level.

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the Device link to display the Device Information panel.

**NOTE:** You can access the version information whether you have read-write ("set") or read-only ("get") access.

## ServerIron Feature List

Table 2.4 on page 2-5 lists the ServerIron software features. For information about a feature, see the page number listed in the right column.

**Table 2.4: ServerIron Software Features**

Feature	See page...
<b>Access and Management Features</b>	
Command-line and web-based management interfaces	2-8
Multiple levels of access control	2-10
RADIUS authentication	2-10
TACACS/TACACS+ authentication	2-10
Access Control Lists (ACLs)	2-11
Dynamic configuration	2-11
Soft reboot (reboot flash image without resetting the system)	2-11
Scheduled system reload	2-11
Telnet	2-11
Trivial File Transfer Protocol (TFTP)	2-12
Simple Network Time Protocol (SNTP)	2-12
Domain Name Server (DNS) resolver	2-12
SNMPv2c	2-12
Remote Monitoring (RMON)	2-13
SNMP alarms and trap log	2-13
SyslogD client	2-13
Ping and trace-route facilities	2-13
Port mirroring	2-14
TCP/UDP port profiles (for globally defining ports)	2-18
Health checks	2-19
<b>Layer 4 Server Load Balancing Features</b>	

**Table 2.4: ServerIron Software Features (Continued)**

<b>Feature</b>	<b>See page...</b>
Application grouping	2-23
Unlimited virtual IP addresses (VIPs)	2-25
SLB Multinetting using Network Address Translation (NAT)	2-15
IP Filters	2-16
Geographically-distributed servers	2-25
Global SLB (GSLB)	2-26
Symmetric Server Load Balancing	2-26
SwitchBack	2-27
Many-to-one TCP/UDP port binding	2-29
HTTP redirect	2-29
Graceful server shutdown and force shutdown option	2-21
Configurable predictor (load-balancing method)	2-23
Health checks	2-19
SNMP traps	2-20
Hot Standby Redundancy	2-21
Transparent VIP and stateless application ports	2-29
<b>Layer 4 Web Switching Features</b>	
URL Switching	2-30
Cookie Switching	2-31
HTTP Header Hashing	2-32
SSL Session ID Switching	2-33
<b>Layer 4 Transparent Cache Switching Features</b>	
Cache Route Optimization (CRO)	2-36
Policy-based Cache Failover (CFO)	2-36
Stateful TCS	2-35
FastCache	2-19
Support for concurrent SLB and TCS	2-14
Proxy Server Cache Load Balancing	2-14
Multinetting using Network Address Translation (NAT)	2-14
IP Filters	2-16
Graceful server shutdown and force shutdown option	2-21
TCP/UDP port profiles (for globally defining ports)	2-18

Table 2.4: ServerIron Software Features (Continued)

Feature	See page...
Health checks	2-19
SNMP traps	2-20
Hot Standby Redundancy	2-14
Layer 4 Firewall Load Balancing Feature	2-37
<b>IP Forwarding Feature</b>	
IP forwarding	2-38
<b>Address Translation Feature</b>	
Network Address Translation (NAT)	2-38
<b>Layer 2 Switching Features</b>	
MAC switching	2-38
Static MAC entries	2-39
Standard Spanning Tree Protocol	2-39
IronSpan STP enhancements	2-40
Trunk groups	2-40
Port-based Virtual LANs (VLANs)	2-40
802.1p VLAN tagging	2-40
MAC filters	2-40
Address-lock filters	2-41
Dynamic Host Configuration Protocol (DHCP) Assist	2-41
IP Multicast Containment	2-41

## Global Network Address Translation

You can configure the ServerIron to perform standard Network Address Translation (NAT). **NAT** enables private IP networks that use nonregistered IP addresses to connect to the Internet. Configure NAT on the Foundry device at the border of an inside network and an outside network (such as the Internet). NAT translates the internal local addresses to globally unique IP addresses before sending packets to the outside network. NAT also allows a more graceful renumbering strategy for organizations that are changing service providers or voluntarily renumbering into Classless Interdomain Routing (CIDR) blocks.

**NOTE:** The standard NAT support described in this section provides IP address translation for hosts attached to private networks on the ServerIron, and is separate from the virtual IP address features provided for Server Load Balancing (SLB). For example, standard NAT is not related to source IP addresses used for multinetting the ServerIron, performing health checks on remote servers, and so on.

Use NAT to translate your private IP addresses into globally unique IP addresses when communicating outside of your network.

For more information, see “Configuring Network Address Translation” on page 14-1.

## Access and Management Features

The following sections describe the access and management features listed in Table 2.4 on page 2-5.

### Management Interfaces

Foundry switches and routers can be managed using the following interfaces:

- Command Line Interface (CLI) – a text-based interface accessible through a direct serial connection or a Telnet session.
- Web management interface – A GUI-based management interface accessible through an HTTP (web browser) connection.

#### Command Line Interface (CLI)

The CLI comes standard on all Foundry switches and routers. The CLI is a text-based operator interface that allows you to configure a system with a PC or terminal without special software.

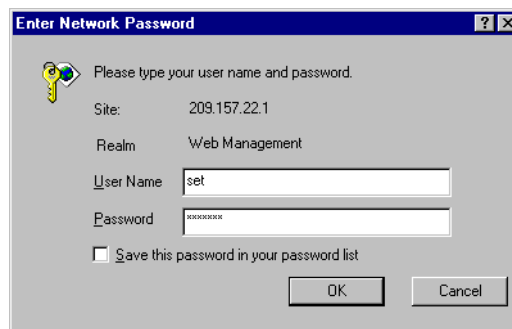
Up to five read-only Telnet sessions can operate concurrently on either a Foundry switch or router. Only one read-write Telnet session is allowed at a time.

#### Web Management Interface

A Web management interface is supported on web browsers Netscape Navigator™ versions 2.0 or later, and Microsoft Internet Explorer™ versions 3.0 or later. No application software is required. The Web management interface comes standard on all switches and routers.

To use the Web management interface, open a web browser and enter the IP address of the Foundry device in the Location or Address field. The web browser contacts the Foundry device and displays a login dialog, as shown in Figure 2.5.

**Figure 2.5** Web Management interface login dialog



---

**NOTE:** If you have configured a greeting banner (using the **banner motd** CLI command), a panel with the greeting is displayed first. Click on the Login link to proceed to the Login dialog.

---

---

**NOTE:** If you are unable to connect with the device through a Web browser due to a proxy problem, it may be necessary to set your Web browser to direct Internet access instead of using a proxy. For information on how to change a proxy setting, refer to the on-line help provided with your Web browser.

---

By default, Web management access is secured using the ServerIron's read-only ("get") and read-write ("set") community strings.

- The default read-only community string is "public". To open a read-only Web management session, enter "get" and "public" for the user name and password.
- Beginning with software release 05.0.00, there is no default read-write community string. Thus, by default, you cannot open a read-write management session using the Web management interface. You first must configure a read-write community string using the CLI. Then you can log on using "set" as the user name and

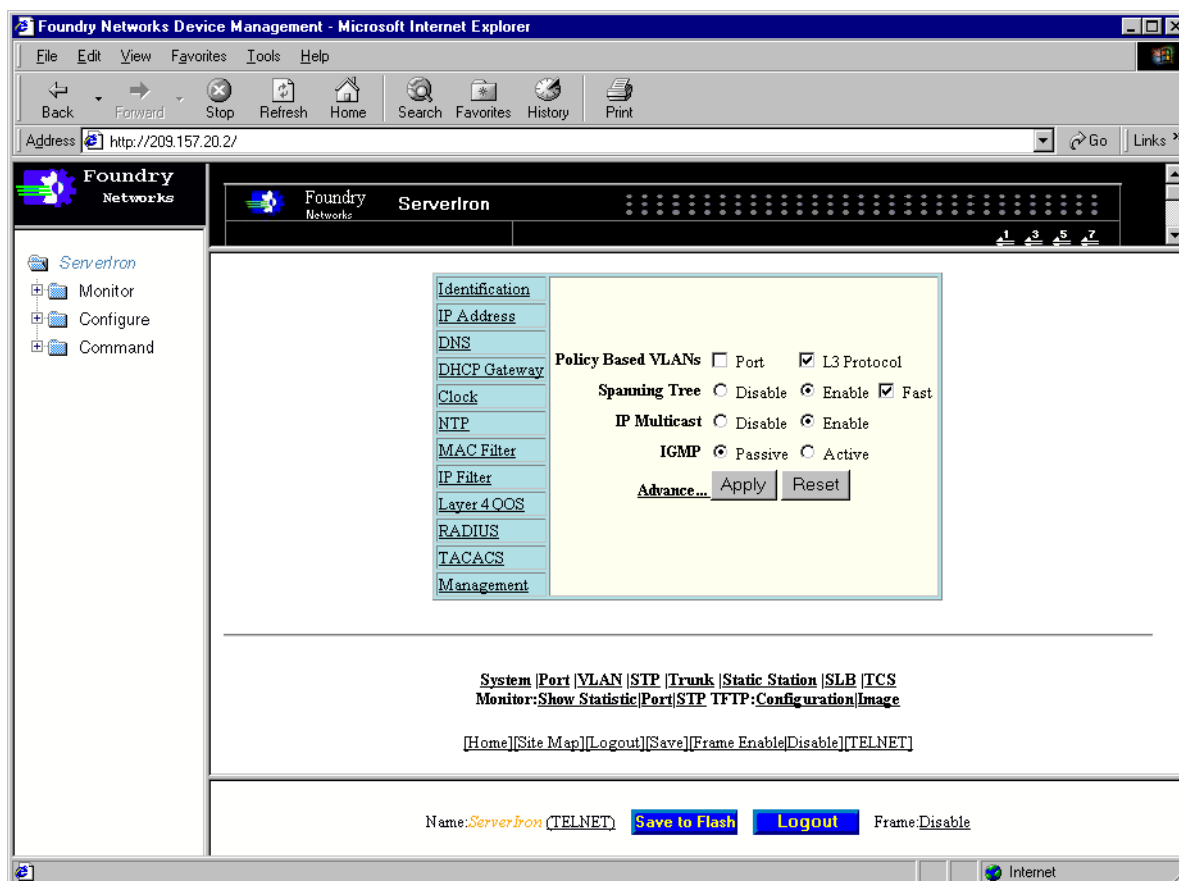


the read-write community string you configure as the password. See “Establishing SNMP Community Strings” on page 3-11.

**NOTE:** As an alternative to the SNMP community strings, you can secure Web management access using local user accounts. See the *Foundry Security Guide*.

Figure 2.6 shows an example of the opening Web management display for a ServerIron.

**Figure 2.6** Example of Web management interface



The configuration and management procedures in this guide include instructions for the Web management interface.

- To display general system information, click on a blank area of the device’s management module. If the front panel display is disabled as shown in this example, click on the object shown in the chassis window. The object contains the product name.
- To display information about a specific port, click on the port on the front panel display. (This option is available only when you enable display of the front panel. See the note below.)
- Click on the links in the left-hand frame or on the bottom of the display to view statistics or to view and change configuration parameters.

---

**NOTE:** The Web management interface automatically refreshes the system information at regular intervals, including the link LEDs for the ports. To streamline performance, display of the front panel is disabled by default. To enable front panel display, select the Configure folder, then the System folder, then the Management folder, and click the [Web Preference](#) link. Select the Enable radio button for Front panel display, then click Apply. Select Reload or Refresh in your browser's tool bar to immediately view the effect of this change.

---

## Multiple Levels of Access Control

Foundry switches and routers provide multiple levels of access to allow system administrators complete configuration control while protecting the system from unauthorized changes.

### CLI Access

The following levels of password protection offer a range of access points for various users within the network:

- **Super user** – Allows a user unlimited access to all levels of the CLI. This level is generally reserved for system administrators within the network. The super user is also the only one who can assign a password access level to another user.
- **Configure port** – Allows a user to configure interface parameters only and to view any show command displays.
- **Read only** – Allows a user to view configuration information. No configuration is allowed at this password level.

---

**NOTE:** You can configure a password for each level of access. In addition, you can configure up to 16 user accounts. See the *Foundry Security Guide*.

---

### Web Management Interface Access

By default, Web management access is secured using the ServerIron's read-only ("get") and read-write ("set") community strings.

- The default read-only community string is "public". To open a read-only Web management session, enter "get" and "public" for the user name and password.
- Beginning with software release 05.0.00, there is no default read-write community string. Thus, by default, you cannot open a read-write management session using the Web management interface. You first must configure a read-write community string using the CLI. Then you can log on using "set" as the user name and the read-write community string you configure as the password. See "Establishing SNMP Community Strings" on page 3-11.

---

**NOTE:** As an alternative to the SNMP community strings, you can secure Web management access using local user accounts. See the *Foundry Security Guide*.

---

### TACACS and TACACS+ Security

You can secure CLI access to the switch or router by configuring the device to consult a Terminal Access Controller Access Control System (TACACS) or TACACS+ server to authenticate user names and passwords. See the *Foundry Security Guide*.

---

**NOTE:** TACACS/TACACS+ authentication is not supported for Web management or IronView access.

---

### RADIUS Security

You can further secure CLI access to the switch or router by configuring the device to consult a RADIUS server to authenticate user names and passwords. You can configure the device to authenticate Telnet logins and Enable access on a separate basis. See the *Foundry Security Guide*.

---

**NOTE:** RADIUS authentication is not supported for Web management or IronView access.

---

## Access Control Lists (ACLs)

**Access control lists (ACLs)** enable you to permit or deny packets based on source and destination IP address, IP protocol information, or TCP or UDP protocol information. You can configure the following types of ACLs:

- Standard – Permits or denies packets based on source IP address. ACL IDs 1 – 99 are for standard ACLs.
- Extended – Permits or denies packets based on source and destination IP address and also based on IP protocol information. ACL IDs 100 – 199 are for extended ACLs.

For syntax information and examples, see the “Using Access Control Lists (ACLs)” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

In addition, you can use ACLs to control CLI, Web, and IronView management access to the device. See the *Foundry Security Guide*.

## Dynamic Configuration

Dynamic configuration enables you to make configuration changes without rebooting the system. Many of the configuration changes you can make to Foundry switches and routers do not require a reboot and take effect immediately. You can make the changes without causing network outages. Most Layer 2 configuration changes are dynamic. All Layer 4-7 configuration changes are dynamic.

For the few Layer 2 parameters that do require a software reload, the individual configuration chapters in this guide and in the *Foundry Switch and Router Installation and Basic Configuration Guide* indicate when a reload is required.

## Soft Reboot

When you upgrade the software image on a Foundry switch or router, you do not need to power down the system to use the new software. You can boot the new software immediately from the primary flash, secondary flash, a TFTP server, or a BootP server.

You also can use this feature to test new versions of flash code before replacing the previous flash image.

For more details on the boot commands and on copying software to and from Foundry switches and routers, see the “Updating Software Images and Configuration Files” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## Scheduled System Reload

Although the dynamic configuration feature allows many parameter changes to take effect immediately without a system reset, other parameters do require a system reset.

To place these parameters into effect, you must save the configuration changes to the configuration file, then reload the system. The management interfaces provide an option to immediately reset the system. Alternatively, you can use the scheduled system reload feature to configure the system to reload its flash code at a specific time (based on the system time counter or SNTP time) or after a specific amount of time has passed.

See the “Updating Software Images and Configuration Files” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## Telnet

As described in “Management Interfaces” on page 2-8, Foundry devices allow you to access the CLI through a Telnet connection. To establish the Telnet connection, you need the following:

- An IP address on the Foundry device. See “Assigning IP Addresses” on page 3-5 for information.
- A third-party terminal emulation application installed on a PC or workstation that has network access to the Foundry device.

## Trivial File Transfer Protocol (TFTP)

All Foundry devices allow you to use TFTP to copy files to and from the flash memory modules on the management module. You can use TFTP to perform the following operations:

- Upgrade boot or flash code.
- Archive boot or flash code or a configuration file on a TFTP server.
- Load the system using flash code and a configuration file stored on a TFTP server. (This occurs as part of the BootP or DHCP process.)

---

**NOTE:** Certain boot upgrades may require you to install new firmware. Contact your reseller or Foundry Networks for information.

---

See the “Updating Software Images and Configuration Files” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide* for more information about using TFTP on Foundry devices.

## Simple Network Time Protocol (SNTP)

Foundry devices can use either of two time and date sources:

- An on-board system clock.
- An external SNTP server. The server can be on the same sub-net or a different sub-net.

If you have access to an SNTP server, Foundry Networks recommends that you use the SNTP server as the time and date source. Using an SNTP server ensures that all devices that use the SNTP server have a consistent time and date. In addition, the settings on the system time counter are not retained across power cycles. The counter has to be reset following each power-up. If the device is configured to reference an SNTP server, the device automatically sets its system time counter according to the SNTP server after a system reset.

Regardless of the time and date source you use, you can configure the time zone of the time and date. You also can enable daylight savings time, which is disabled by default.

See the “Configuring Basic Features” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## Domain Name Server (DNS) Resolver

The DNS Resolver feature allows you to use just a host name rather than a fully-qualified domain name when you use Telnet, ping, and trace-route commands. To configure the feature, you specify the domain name, then specify the IP addresses of up to four DNS servers that have authority for the domain.

For example, if you define the domain “newyork.com” on a Foundry device, you can initiate a ping to a host on that domain by specifying only the host name in the command. You do not need to specify the host’s entire domain name.

As an example, here are two CLI commands. The first command uses only the host name. The second command uses the fully-qualified domain name for the host.

```
ServerIron# ping nyc01
```

```
ServerIron# ping nyc01.newyork.com
```

See the “Configuring Basic Features” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## SNMPv2c Support

Foundry devices support SNMPv2c, including support for GetBulk requests. The SNMPv2c support is enabled by default and cannot be disabled. Thus, you do not need to perform any configuration on the device to use the feature. SNMP V1 also is enabled by default.

---

**NOTE:** You can disable SNMP access to the device if needed. See the *Foundry Security Guide*.

---

To use this enhancement, you need an SNMP management application that is capable of sending GetBulk requests. See the documentation for your application for more information.

---

**NOTE:** The SNMPv2c support does not include support of SNMPv2c traps. Also, IronView/UNIX does not support SNMPv2c.

---

## Remote Monitoring (RMON) Statistics

All Foundry devices include an RMON agent that supports the following groups. The group numbers come from the RMON specification (RFC 1757).

- Statistics (RMON Group 1) – Current packet and error statistics for each port.
- History (RMON Group 2) – Samples of packet and error statistics captured at regular intervals. You can configure the sample rate and the number of "buckets" in DRAM for storing the samples.
- Alarms (RMON Group 3) – A list of alarm events, which indicate that a threshold level for a specific part of the device has been exceeded. You can select the system elements you want RMON to monitor and the thresholds for triggering the alarms.
- Events (RMON Group 9) – A log of system events (such as port-state change to up or down, and so on) and alarms. RMON Group 9 also specifies the action to be taken if an alarm threshold is exceeded.

See "Network Monitoring" on page B-1 for more information.

## Syslog Logging

In addition to the event and alarm logs provided by RMON, Foundry devices contain a Syslog agent that can write log messages to a local buffer and optionally to a third-party SyslogD server. The Syslog feature can write messages at the following severity levels:

- Emergencies
- Alerts
- Critical
- Errors
- Warnings
- Notifications
- Informational
- Debugging

The device automatically writes the Syslog messages to a local buffer. If you specify the IP address or name of a SyslogD server, the device also writes the messages to the SyslogD server. The default facility for messages written to the server is "user". You can change the facility if needed. You also can change the number of entries that can be stored in the local buffer. The default is 50. The ServerIron does not have a limit to the number of messages that can be logged on a remote SyslogD server.

---

**NOTE:** You can specify only one facility.

---

For more information about logging, see the "Configuring Basic Features" chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

For a list of the Syslog messages, see the "show logging" section in the "Show Commands" chapter of the *Foundry ServerIron Command Line Interface Reference*.

## Ping and Trace-Route Facilities

After you configure an IP address for the device, you can test the device's network connections using the following facilities:

- Ping – You can send a test packet to a host's IP address or host name. If the packet reaches the host, the host generally sends a reply packet to let you know the host received your ping. If the host does not reply within a specified interval, the Foundry device re-attempts the ping up to a specified number of times.
- Trace-route – On Foundry switches and Layer 3 Switches, you can trace the IP path to a host. The trace-route feature displays a list of all the intervening router hops the trace-route request traversed to reach the host.

See "Testing Connectivity" on page 3-10.

## Port Mirroring

The mirror port feature lets you connect a protocol analyzer to a port on a Foundry device to observe the traffic flowing into and out of another port on the same device. To use this feature, you specify the port you want to monitor and the port into which you are plugging the protocol analyzer.

---

**NOTE:** Only one mirror port can be active on a switch or router at a time. By default, no mirror port is assigned.

---

See the "Configuring Basic Features" chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## Basic Layer 3 Services

The ServerIron does not contain Layer 3 routing protocols and thus does not require routing configuration. However, the ServerIron does support the following Layer 3 services:

- Address Resolution Protocol (ARP)
- Ping for Layer 3 health checking of servers using Internet Control Message Protocol (ICMP) echo packets
- Multiple source IP addresses and network address translation (NAT) for multinetted environments

The first two services are automatic and require no configuration. You configure NAT support, if needed, by adding additional IP addresses to the ServerIron and in some cases enabling source NAT or destination NAT. This guide describes how to configure the ServerIron for NAT where applicable.

## Proxy Server Cache Load Balancing

This feature configures the ServerIron to perform TCS for clients whose web browsers are configured to use a proxy. By configuring the ServerIron to serve these clients, you relieve clients from the need to modify their browser configurations. Clients whose browsers do not use a proxy are still served transparently by the ServerIron's TCS mechanism. Clients who do use a proxy also are served transparently, by a virtual IP address (VIP) that you configure on the ServerIron with the same IP address as the proxy.

See "Proxy Server Cache Load Balancing" on page 10-42 for more information.

## Selectable Quality of Service (QoS)

The ServerIron provides the following types of selectable Quality of Service (QoS):

- Layer 2 flow control – All Foundry devices provide support for the Full Duplex Flow Control specification, 802.3x.
- Layer 2 802.1p/802.1q QoS support – Provides benefits beyond the local switch or router by supporting and recognizing standard-based virtual LAN (VLAN) tagging, in addition to providing support for flow control and port priority.
- Policy – Layer 2 packet-based priority for individual ports, VLANs, and static MAC entries. You can apply a QoS priority of "normal" or "high" to these items. In addition, you can apply the priority "cache" or "fw", which enables TCS or firewall load balancing.
- Layer 4 session packet-based priority.

All these items are in the normal queue by default. You can place items in the high priority queue to ensure that their traffic is forwarded before normal traffic.

---

**NOTE:** You enable TCS by setting the device's QoS policy to "cache". You enable firewall load balancing by setting the QoS policy to "fw". The other values are "normal" and "high". See "Enabling TCS" on page 10-4 and the "Configuring Basic and IronClad FWLB" chapter in the *Foundry ServerIron Firewall Load Balancing Guide*.

---

## SLB Multinetting Using Network Address Translation (NAT)

The ServerIron can support all the variations of network address translation (NAT) listed in Table 2.5 on page 2-15. The NAT support enables the ServerIron to operate in a multi-netted environment.

---

**NOTE:** The standard NAT support described in "Multiple Levels of Access Control" on page 2-10 provides IP address translation for hosts attached to private networks on the ServerIron, and is separate from the virtual IP address features provided for Server Load Balancing (SLB). For example, standard NAT is not related to source IP addresses used for multinetting the ServerIron, performing health checks on remote servers, and so on.

---

Address translation applies only to SLB. The default NAT behavior for SLB is as follows:

- For ServerIron->real server packets:
  - Destination – Translate address from virtual IP address (VIP) into real server's IP address.
  - Source – Leave client's IP address unchanged. The MAC address is changed to the ServerIron's MAC address.
- For ServerIron->client packets:
  - Destination – Leave client's IP address unchanged.
  - Source – Translate real server IP address into VIP address.

The ServerIron always translates the MAC address in responses from a real server into the MAC address of the virtual IP address (VIP) before sending the response to the client. Thus, the client receives a response that contains the source IP address and MAC address of the VIP.

This behavior assumes that the ServerIron and the real servers are all on the same sub-net and have direct Layer 2 connectivity. However, you are not limited to this topology. You can easily configure the ServerIron to operate in a multi-netted environment in which the ServerIron and the real servers are in different sub-nets.

In addition to the IP management address, the ServerIron can have up to eight additional IP addresses for use with source NAT. When the network topology requires the ServerIron to translate a packet's source IP address into one of the ServerIron's source IP addresses, the ServerIron can use one of the source IP addresses you configure. You can configure source IP addresses on a global basis (for the entire device). See the application examples in "SLB Application Examples" on page 6-96 for more information.

**Table 2.5: ServerIron NAT Support**

Translation		Direction	Application
Source IP Address	Destination IP Address		
	X	ServerIron->real server	Destination – Translate virtual IP address known by client into real server address.
X		ServerIron->client	Source – Translate real server IP address into virtual IP address known by client.

**Table 2.5: ServerIron NAT Support (Continued)**

Translation		Direction	Application
Source IP Address	Destination IP Address		
X	X	ServerIron->real server	<p>In multi-netted environment:</p> <ul style="list-style-type: none"> <li>Destination – Translate virtual IP address known by client into real server address.</li> <li>Source – Translate client IP address into source IP address in the same sub-net as the real server if possible. (Source IP address is defined on the ServerIron.)</li> </ul> <p>When sending client request to remote real server:</p> <ul style="list-style-type: none"> <li>Destination – Translate virtual IP address known by client into real server address.</li> <li>Source – Translate client IP address into source IP address defined on the ServerIron. This ensures that server response comes back to ServerIron instead of directly to client.</li> </ul>
X	X	ServerIron->client	<p>In multi-netted environment:</p> <ul style="list-style-type: none"> <li>Source – Translate real server address into virtual IP address known by client.</li> <li>Destination – Translate ServerIron source IP address back into client IP address.</li> </ul> <p>When receiving response from remote server:</p> <ul style="list-style-type: none"> <li>Source – Translate real server address into virtual IP address known by client.</li> <li>Destination – Translate ServerIron source IP address into client IP address.</li> </ul>

## IP Filters

You can use IP filters or Access Control Lists (ACLs) to selectively control SLB and TCS traffic. The filters or ACLs can match on source and destination IP address, network mask, and TCP/UDP port information.

This section describes IP filters. For standard IP ACL configuration information, see the “Configuring Standard ACLs” section in the “Using Access Control Lists (ACLs)” chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide*.

All filters and ACLs are dynamic; they take place immediately for new connections and do not require a reboot of the ServerIron. New filters or ACLs do not affect existing connections.

Each filter or ACL provides one of the following actions:

- Permit
  - For SLB, permits access to a virtual server (identified by VIP) or to a specific TCP/UDP port on the virtual server.
  - For TCS, permits redirection of a client request to a cache server.



- Deny
  - For SLB, denies access to a virtual server (identified by VIP) or to a specific TCP/UDP port on the virtual server. The packet is dropped.
  - For TCS, denies access to the cache server and instead sends the request out to the Internet. The packet is not dropped.

By default, no filters or ACLs are configured on the ServerIron. All packets are implicitly permitted. However, as soon as you add a filter or ACL, all packets that do not match the filter or ACL are implicitly denied. This behavior ensures tighter control in filtered environments. To change this behavior so that all packets that do not match a filter are permitted instead of denied, configure the last filter (1024) or ACL to permit any traffic.

---

**NOTE:** To filter on Layer 2 traffic, you can configure Layer 2 MAC filters. See “MAC Filters” on page 2-40.

---

This document does not describe how to configure ACLs. For ACL configuration information, see the “Using Access Control Lists (ACLs)” chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## TCS Uses of Filters

You can use filters in TCS to control the following:

- Whether a specific request is sent to a cache server or forwarded to the Internet
- Whether content from specific sites is cached. You can even use policy-based cache switching to determine which cache servers receive content from specific sites.

---

**NOTE:** TCS filters never drop packets. Accept filters send packets to a cache server. Deny filters send packets to the Internet.

---

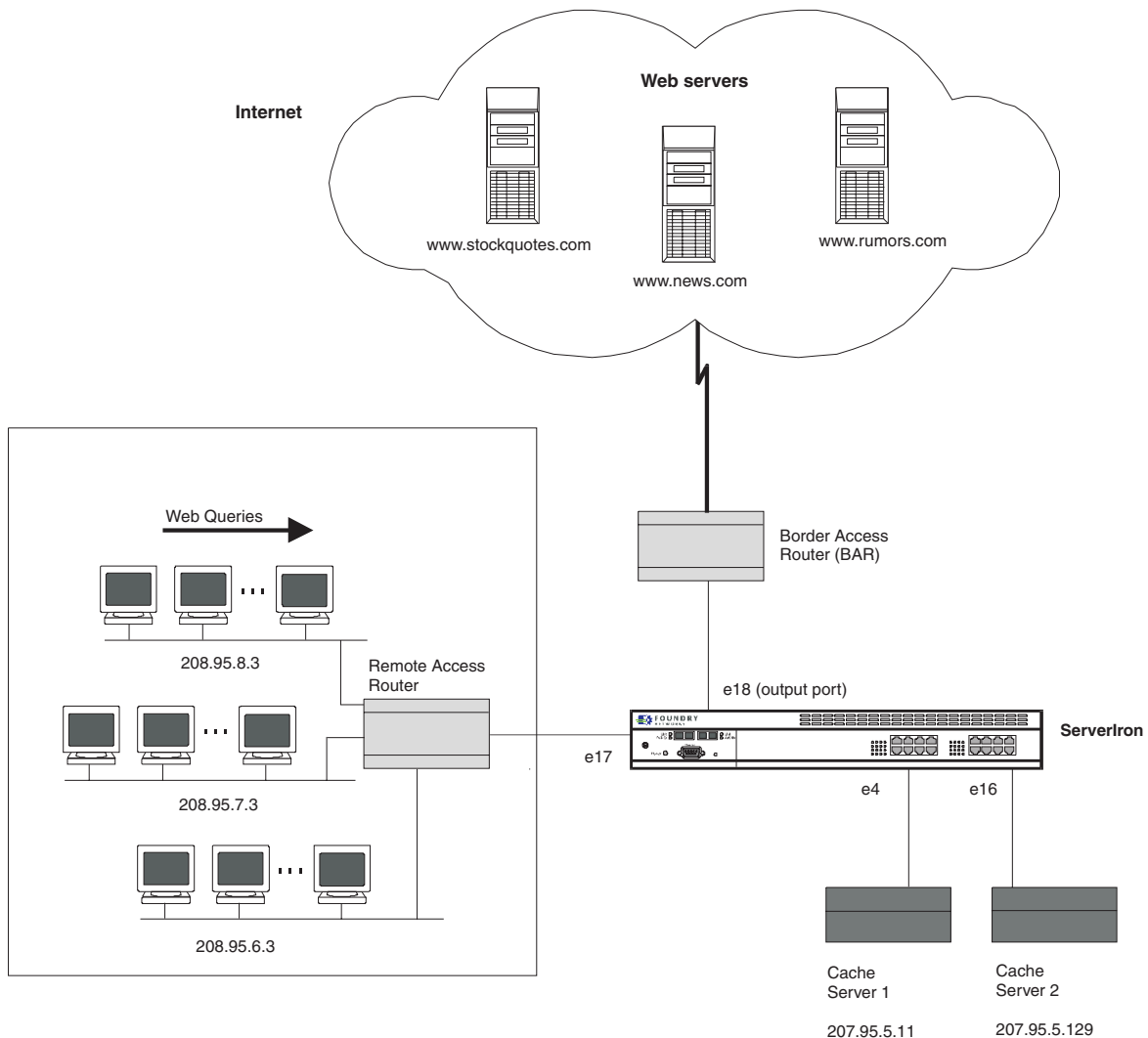
If you do not define any filters, the default action is permit. For TCS, the default action redirects all traffic to cache servers. However, when you define a filter, the ServerIron changes the default action to deny to ensure tighter control. If you still want the default action to be permit, you can define the last filter (1024) to permit all traffic.

Filters apply only to new connections. New filters do not affect existing connections.

You can turn off web caching for a certain range of source or destination addresses to allow filtering on an address basis using IP filters.

For the example in Figure 2.7, you can use a deny filter to force all web queries from a specific sub-net (such as 208.95.6.3) to go directly to the Internet (the normal destination) rather than be redirected to cache servers. In this case, you define a filter by the source address.

**Figure 2.7 Using IP filters to bias traffic away from caching**



### Policy-Based Cache Switching

The ServerIron TCS software allows you to configure IP filters to selectively cache or not cache content from specific web sites on specific cache servers. For example, suppose some of your cache servers come preconfigured with specific web pages and you want all updates to those pages to go only to the preconfigured caches. In this case, you can use policy-based cache switching along with IP filters to configure the ServerIron to send the content only to the specified cache servers.

You also can configure IP filters to prevent specific web sites from being cached on specific cache servers or all cache servers.

See "Policy-Based Caching" on page 10-37 for more information about this feature.

## Port Profiles

A **port profile** is a set of attributes that globally define a TCP/UDP port. Once defined, the port has the same attributes on all the real and virtual servers that use the port.

A port profile consists of the following attributes:

- Port type (TCP or UDP) – this attribute applies only to ports for which the ServerIron does not already know

the type. For example, if a real server uses port 8080 for HTTP (a TCP port), you can globally identify 8080 as a TCP port. The ServerIron assumes that ports for which it does not know the type are UDP ports.

- Keepalive interval and retries – the number of seconds between health checks and the number of times the ServerIron re-attempts a health check to which the server does not respond
- Keepalive state – whether the ServerIron's health check for the port is enabled or disabled
- TCP or UDP age – the number of minutes a TCP or UDP server connection can remain inactive before the ServerIron times out the session. This parameter is set globally for all TCP or UDP ports but you can override the global setting for an individual port by changing that port's profile.

Port profiles enable you to characterize a port globally, at the global system level. For example, if many of your real servers use TCP port 80 (the well-known number for HTTP) and you want to change the keepalive interval for the port, you can do so globally. You do not need to change the value individually on each real server, at the Real Server level.

The ServerIron knows the port types of a some well-known port numbers. If you are using a port number for which the ServerIron does not know the port type, you can specify whether the port is TCP or UDP and configure its keepalive values globally. You do not need to define the port on every server.

Health checks are described in the following section. For information about configuring port profile parameters, see "Configuring Port and Health Check Parameters" on page 12-1.

## Health Checking

The ServerIron provides Layer 3, Layer 4, and Layer 7 health checks for servers and the individual applications on the servers.

- Layer 3 health checks – The ServerIron uses IP pings to determine whether a server can be reached.
- Layer 4 health checks – The ServerIron sends TCP or UDP requests to individual TCP or UDP ports to determine whether the ports are healthy.
- Layer 7 health checks – For certain TCP and UDP application ports, the ServerIron can send application-specific health checks to determine the health of the application. For example, the ServerIron can send user-configurable HTTP requests to real servers to assess the health of the servers.

For more information, see "Configuring Port and Health Check Parameters" on page 12-1.

## ICMP Message Feature for HTTP

By default, if a client requests a TCP/UDP port that is not available, the ServerIron does not send an ICMP "Destination Unreachable" message to the client. For HTTP traffic, you can configure the ServerIron to send such a message to the client by enabling the ICMP message feature. When this feature is enabled, the ServerIron sends an ICMP "Destination Unreachable" message to the client if the requested port either is not configured on any of the real servers or is unavailable because all the servers configured with the requested port are busy or down.

See "ICMP Unreachable Messages" on page 6-29 for information about enabling this feature.

## FastCache

In typical TCS configurations, the ServerIron uses cache responses that flow back through the ServerIron as a means to determine the health of the cache server.

When the ServerIron receives cache responses to client requests sent to the cache by the ServerIron, the ServerIron knows that the cache server is healthy. However, if the cache server does not respond to client requests, the ServerIron does not receive the responses from the cache server. Therefore, the ServerIron determines that the cache server is down and stops sending client requests to the cache server.

Some configurations might require responses from a cache server to select a path that does not return through the ServerIron. For example, if a cache server supports only one default path and that path is to a gateway router, not to the ServerIron, the cache server might send responses to the clients through the default gateway instead of

through the ServerIron. In this case, the ServerIron assumes that the cache server has stopped responding even though the cache server is still working normally.

You can override health checking on an individual server basis by enabling FastCache. This feature allows the ServerIron to continue using a cache server even if the server does not send responses to client requests back through the ServerIron. When you enable FastCache on a cache server, the ServerIron continues to send client requests to the cache server even if the server does not respond through the ServerIron.

---

**NOTE:** FastCache (used in TCS) and SwitchBack (used in SLB) are different features. See “SwitchBack” on page 2-27 for information about SwitchBack.

---

## Simple Network Management Protocol (SNMP) Traps

The ServerIron supports the following traps.

- SNMP Authentication – Indicates a failed attempt to access the device through SNMP using an invalid SNMP community string.
- Power Supply – Indicates a power supply failure.
- Fan – Indicates a fan failure.
- Cold Start – Indicates a restart from a powered down state.
- Link Up – Indicates that a port link has come up.
- Link Down – Indicates that a port link has gone down.
- Bridge New Root – Indicates a spanning-tree change.
- Bridge Topology Change – Indicates a spanning-tree change.
- Lock Address Violation – Indicates that a locked port received a packet for a MAC address that is not allowed access to that port. (See “Address-Lock Filters” on page 2-41 for more information about port locking.)
- Maximum Session – Indicates that the maximum number of sessions has been reached. A session is either a send or receive link between the ServerIron and a real server. Two sessions make a two-way connection between the ServerIron and a server.
- TCP SYN Limit – Indicates that the maximum TCP SYN rate has been reached on a real server.
- Real Server Max Connection – Indicates that a real server has reached the maximum number of connections the ServerIron is configured to allow on that server. A connection represents both the receive and send sessions.
- Real Server Up – Indicates that a real server has come up.
- Real Server Down – Indicates that a real server has gone down.
- Real Server Port Up – Indicates that a port on a real server has come up.
- Real Server Port Down – Indicates that a port on a real server has gone down.
- Cache Server Up – Indicates that a cache server has come up.
- Cache Server Down – Indicates that a cache server has gone down.
- Cache Server Port Up – Indicates that a TCP port on a cache server has come up.
- Cache Server Port Down – Indicates that a TCP port on a cache server has gone down.
- Switch Standby – Indicates that an SLB switch fail-over has occurred, and the active switch is down.
- Switch Active – Indicates that the standby switch is active.

See the *Foundry Switch and Router Installation and Basic Configuration Guide* for information about specifying an SNMP trap server and viewing ServerIron traps. All traps are enabled by default. Also see the *Foundry Switch and Router Installation and Basic Configuration Guide* for information about disabling traps.

## Graceful or Forced Server Shutdown

SLB and TCS allow the graceful shutdown of servers and services. By default, when a service is disabled or deleted, the ServerIron does not send new connections the real servers for that service. However, the ServerIron does allow existing connections to complete normally, however long that may take.

You can use the **force shutdown** option to force the existing connections to be terminated within two minutes.

---

**NOTE:** If you disable or delete a service, do not enter an additional command to reverse the command you used to disable or delete the service, while the server is in graceful shutdown.

---

See the section on shutting down a server in “Configuring Server Load Balancing” on page 6-1 or “Configuring Transparent Cache Switching” on page 10-1 for more information.

## Hot Standby Redundancy

Hot standby redundancy allows you to configure two Foundry ServerIrons to serve as a redundant pair. One of the ServerIrons is the active ServerIron and the other remains in standby mode. If the active ServerIron fails, the standby ServerIron assumes the duties of the failed ServerIron and becomes the new active ServerIron.

When switches are configured as backups, one switch serves as the primary or active switch, and the other serves as the secondary or standby switch. The standby switch becomes active only if the primary switch fails due to loss of power or loss of data path.

Devices configured for Hot Standby Redundancy share a MAC address. When you configure devices for Hot Standby, you use the MAC address of the port on the active device that connects to the standby device as the MAC address for the redundant pair. Thus, clients and routers continue to access the same ServerIron MAC address without service interruption, unaware that a device has failed.

See “Configuring Hot Standby Redundancy” on page 5-1 for more information.

---

**NOTE:** For SLB configurations, you can implement the Symmetric SLB feature as an alternative to Hot Standby. Symmetric SLB actively uses both ServerIrons for load balancing, while each ServerIron provides a backup for virtual IP addresses (VIPs) serviced by the other ServerIron. See “Configuring Symmetric SLB and SwitchBack” on page 7-1 for information.

---

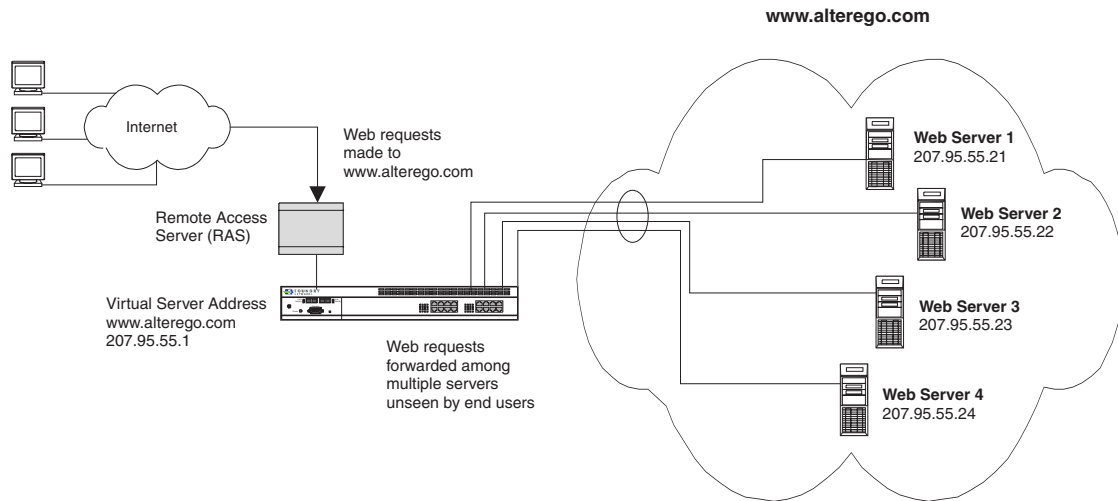
## Server Load Balancing (SLB) Features

Server Load Balancing (SLB) maps one logical (virtual) server connection to multiple physical (real) servers. Thus, a single IP address (virtual server IP address) can serve as the connection point for multiple TCP/UDP services such as HTTP, FTP or Telnet rather than each of the services requiring a different IP address for each service. These services can be located on a single server or across multiple servers.

In Figure 2.8, a company establishes a web site with the URL of [www.alterego.com](http://www.alterego.com). The web site is mapped to the virtual IP address 207.95.55.1, defined on a ServerIron. All inquiries made to that web site by users on the Internet or the company's Intranet use either the URL or virtual IP address to reach the company's web site.

Once these inquiries are received at the company site, the requests are handled by one of four separate physical (real) web servers that the system administrator has mapped to the virtual IP address. The addresses of the four physical (real) web servers are unknown and unseen to those users who send the inquiries. The only address the users ever see for the web site is the virtual IP address.

**Figure 2.8 Single virtual IP address mapped to multiple real servers**



## Value of SLB

SLB provides numerous benefits that ease overall administration of TCP/UDP applications on servers as well as increase their performance and reliability.

In the previous example, Figure 2.8, the system administrator has greater flexibility in managing server resources for this application. When you use a ServerIron, you can add or remove the physical (real) servers to handle changing traffic requirements without disrupting service to the end users. The end users continue to access the virtual IP address configured on the ServerIron and are not aware of added or removed real servers that underlay the virtual IP address.

SLB also enhances server security because the real servers' IP addresses are never broadcast. The ServerIron sends and responds to ARPs with the virtual IP address, not the actual IP addresses of the real servers.

In addition to offering increased control over server resources and greater security within the network, SLB provides increased reliability of the server resources by providing support for both switch and server redundancy.

For more details on switch redundancy, see "Basic Layer 3 Services" on page 2-14.

## How SLB Works

A Foundry ServerIron or switch running SLB software establishes a virtual server that acts as a front-end to physical servers, distributing user service requests among active real servers. SLB packet processing is based on the Network Address Translation (NAT) method. Packets received by the virtual server IP address are translated into the real physical IP address based on the configured distribution metric (for example, "round robin") and sent to a real server. Packets returned by the real server for the end user are translated by SLB so that the source address is that of the virtual server instead of the real server.

NAT translation is performed for both directions of the traffic flow. Converting virtual services to real services requires IP and TCP checksum modifications.

Port translation is not performed for any virtual port that is bound to a default virtual port.

### Slow-Start Mechanism

When the ServerIron begins sending client requests to a real server that has recently gone online, it allows the server to ramp up by using the **slow-start mechanism**. The slow-start mechanism allows a server (or a port on the server) to handle a limited number of connections at first and then gradually handle an increasing number of connections until the maximum is reached.

The ServerIron uses two kinds of slow-start mechanisms:

- The non-configurable **server slow-start mechanism** applies to a real server that has just gone online

- The configurable **port slow-start mechanism** applies to individual TCP application ports that have just been activated on a real server

See “Configuring the Slow-Start Mechanism” on page 12-62 for more information.

## Configurable Load-Balancing Predictor

The ServerIron uses a parameter called the **predictor** to determine how to balance the client load across servers. You can globally configure SLB to use one of the following predictors:

- Least connections – The ServerIron sends the request to the real server that currently has the fewest active connections with clients.
- Least sessions – The ServerIron sends the request to the real server that currently has the fewest session table entries.
- Round-robin – The ServerIron sends the request to each server in rotation, regardless of how many connections or sessions each server has.
- Weighted – The ServerIron uses the weights you assign to the real servers to select a real server. The weights are based on the number of session table entries the ServerIron has for each server.
- Response time only – The ServerIron selects the real server with the fastest response time.
- Least connection and server response time weights – The ServerIron compares a combination of a real server’s least-connections weight and server response time weight to the same values for the other real servers.
- Least local connections (ServerIron 400 or ServerIron 800 only) – On an individual WSM CPU basis, the ServerIron selects the real server with the fewest active connections with clients. The predictor selects the real server that has the least number of connections created by the local WSM CPU. The local WSM CPU is the CPU that is managing the chassis slot connected to the real server.
- Least local sessions (ServerIron 400 or ServerIron 800 only) – On an individual WSM CPU basis, the ServerIron selects the server that has the fewest active session on the WSM CPU attached to the real server. The number of sessions is updated when session entries are deleted.

You also can override the default for individual virtual servers.

## Configurable Application Grouping

By default, the ServerIron uses the predictor (load-balancing method) you configure for each new request from a client to a virtual server. This works well for many situations. However, for some web implementations, it is desirable or even required to have the client continue to access the same real server for subsequent requests.

You can configure the ServerIron to ensure that a client that accesses certain TCP/UDP ports on a VIP always goes to the same real server. For example, you might want to configure the TCP/UDP ports related to an interactive web site so that when a client begins a session on the site, subsequent requests from the client go to the same real server. As another example, you might want the real server to be able to arbitrarily assign open TCP/UDP sessions with the client using ports dynamically allocated by the real server.

Application grouping parameters apply to virtual servers. When you configure a virtual server, you specify the TCP/UDP ports on that virtual server. When you apply application grouping to a TCP/UDP port on a virtual server, the application grouping overrides the predictor. The ServerIron allows you to configure the following application grouping methods on an individual virtual-server basis:

- Sticky connections – When you add a TCP/UDP port to a virtual server, if you specify that the port is “sticky”, a client request for that port always goes to the same real server unless the sticky age timer has expired. The sticky age timer specifies how long the connection remains “sticky” (always goes to the same real server) and is reset each time a new client request goes to the sticky port. Once the sticky timer expires, the ServerIron uses the predictor (load-balancing metric) you have configured to select a real server for requests for a port.
- Configurable TCP/UDP application groups – You can group a “primary” TCP/UDP port with up to four additional TCP/UDP ports. After the ServerIron sends a client request for the primary port to a real server, subsequent requests from the client for ports grouped with the primary port go to the same real server.

- **Concurrent connections** – When you configure a TCP/UDP port in a virtual server to support concurrent connections, the real server can open additional ("concurrent") TCP/UDP sessions with the client using arbitrary TCP/UDP port numbers. Although the concurrent connections feature is similar to the application group feature, application groups apply to specific TCP/UDP ports that you configure on the virtual server. Concurrent connections enables the real server to arbitrarily determine the TCP/UDP ports and assign them to the client.

---

**NOTE:** For servers that use passive FTP, configure the FTP ports to be both sticky and concurrent.

---

### Sticky Connections

When a service request by a client mandates a series of sequential TCP/UDP port connections to be served by the same real server, you can enable a sticky connection on that TCP/UDP virtual server port. For example, if a user is accessing dynamically generated pages, the client must consistently attach to the same server; otherwise, the state information is lost. By default, the sticky parameter is disabled for virtual service ports, except for Secure Socket Layer (SSL).

### Configurable TCP/UDP Application Groups

Application groups enhance the sticky connections method by allowing you to group up to four TCP/UDP ports with another, "primary" TCP/UDP port. When the ServerIron sends a client request for the primary port to a real server, requests from the same client for a port that is grouped with the primary port also go to the same real server. The application group method, like the sticky method, is governed by the sticky age.

The ServerIron automatically sends requests for the grouped ports to the same real server as the "primary" port as long as the sticky timer has not expired. You must define all the ports in an application group individually in the VIP and you must configure all of them to be sticky.

See "TCP/UDP Application Groups" on page 6-104 for an example of this feature.

### Concurrent Connections

The concurrent connection option is similar to the sticky option. However, instead of supporting sequential connections to the same server, the concurrent connection option supports both a primary connection and secondary connections. The connections are supported at the same time.

The primary connection is the controlling connection and dictates the resource, such as a server, to which subsequent or secondary connections are made.

Once the controlling connection is established, the server dynamically assigns a TCP/UDP port to which the client should open subsequent or secondary connections. Subsequent connections from that client are accepted on the server as long as the controlling connection is still active.

Figure 2.9 shows an example of a concurrent connection. A client initiates a session request to the NETPERF application supported on servers S1, S2, and S3. When the SLB switch receives the request, the switch forwards the request to server S2. This forms the primary connection. Then S2 dynamically assigns a port, 10000, to the application and forwards it to the client.

---

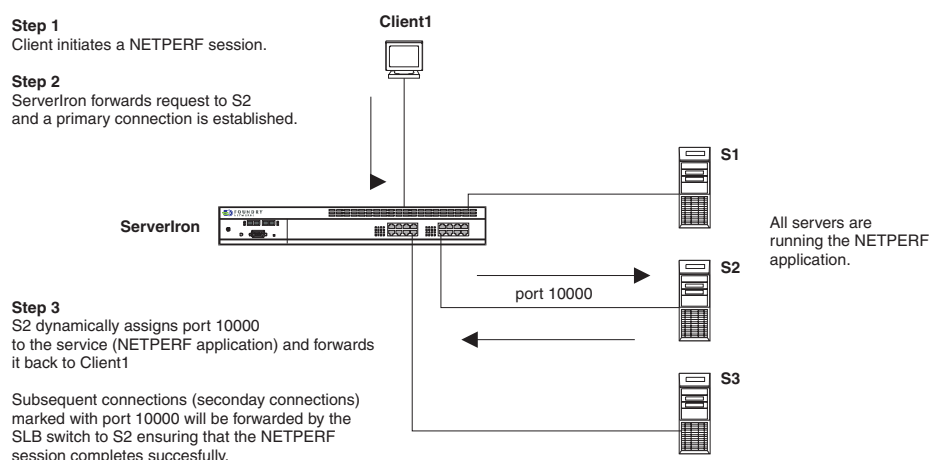
**NOTE:** The method the server uses to communicate the dynamic port to the client is application specific.

---

The ServerIron switches all subsequent connections to S2 to ensure that the NETPERF session completes successfully.

By default, the concurrent property of a virtual TCP/UDP service port is enabled except for FTP, Telnet, TFTP, HTTP, and SSL ports.



**Figure 2.9** Concurrent connections in operation on an SLB network

## Unlimited Virtual IP Addresses (VIPs)

If your real web servers have many IP addresses, you can easily create a separate virtual IP address (VIP) for each real IP address without individually configuring each VIP. To do so, configure a host range. A host range is a block of contiguous IP addresses.

To configure a host range, you add a VIP and specify how many hosts are in the range. The ServerIron automatically configures a separate VIP for each host in the range. When you bind the base VIP to the real servers, the ServerIron associates the VIP with the first real IP address on the server. Subsequent VIPs in the host range are associated with subsequent real IP addresses on the server. The association is based on the offset from the base VIP. When a client requests an address in the VIP range, the ServerIron automatically maps the VIP to a real IP address on a real server based on the address's offset from the base VIP address.

For example, if you define a range using the base VIP 209.157.22.1 and a host range of 10, the ServerIron maps VIPs 209.157.22.1 – 209.157.22.10 to a range of ten addresses on each real server. If a client requests VIP 209.157.22.3 (two from the base VIP number), the ServerIron sends the request to an IP address that is two higher than the start of the corresponding range on a real server.

You can configure up to 256 virtual servers, each with a host range of 256 addresses, for a total of up to 64,000 VIPs.

**NOTE:** To use this feature, the IP address range on the real server must be contiguous. If the range contains gaps (addresses in use by other hosts), specify separate ranges on the virtual server to work around the gaps.

## Geographically-Distributed Servers

The servers in your SLB configurations do not need to have Layer 2 connectivity to the ServerIron. In fact, they do not need to be in the same LAN or Intranet as the ServerIron at all. Using the NAT support described in “SLB Multinetting Using Network Address Translation (NAT)” on page 2-15, you can configure the ServerIron to use geographically-distributed servers.

In a typical configuration, the ServerIron uses geographically-distributed servers as failovers if all the local servers become unavailable. When you configure a real server, you indicate whether the server is local or remote. If the server is remote, the ServerIron does not include the server in its predictor (load-balancing metric). The remote server can be the IP address of a real server or even a virtual IP address configured on another ServerIron. For information about the predictor, see “Configurable Load-Balancing Predictor” on page 2-23.

Servers that are locally attached to the ServerIron (not separated by one or more router hops) are local servers. Servers that are one or more router hops away from the ServerIron are remote servers.

---

**NOTE:** You can configure the ServerIron to include remote servers when load balancing traffic, instead of using the remote servers only as failovers. See “Primary and Backup Servers” on page 6-53.

---

## Global Server Load Balancing (GSLB)

Global Server Load Balancing (GSLB) enables a ServerIron to add intelligence to authoritative Domain Name Servers (DNSs) by serving as a proxy to the servers. As a DNS proxy, the GSLB ServerIron evaluates the server IP addresses in the DNS replies from the DNS for which the ServerIron is a proxy. Based on the results of the evaluation, the GSLB ServerIron can change the order of the addresses in the reply so that the “best” host address for the client is on top.

You can configure a ServerIron to provide GSLB for other ServerIrons. In this case, each of the other ServerIrons is a site ServerIron providing SLB for a local server farm. The GSLB ServerIron uses a policy to select the best site and if necessary modifies the DNS reply to the client accordingly.

For more information, see “Configuring Global Server Load Balancing” on page 9-1.

## Symmetric Server Load Balancing

Symmetric Server Load Balancing (SLB) allows you to use multiple ServerIrons to simultaneously load balance VIP traffic and provide hot standbys for one another’s VIPs. In addition to their roles as mutual backups, each ServerIron actively provides Layer 4 SLB (and TCS, if configured) services. As a result, you get the fault protection of a hot standby configuration while doubling connection capacity.

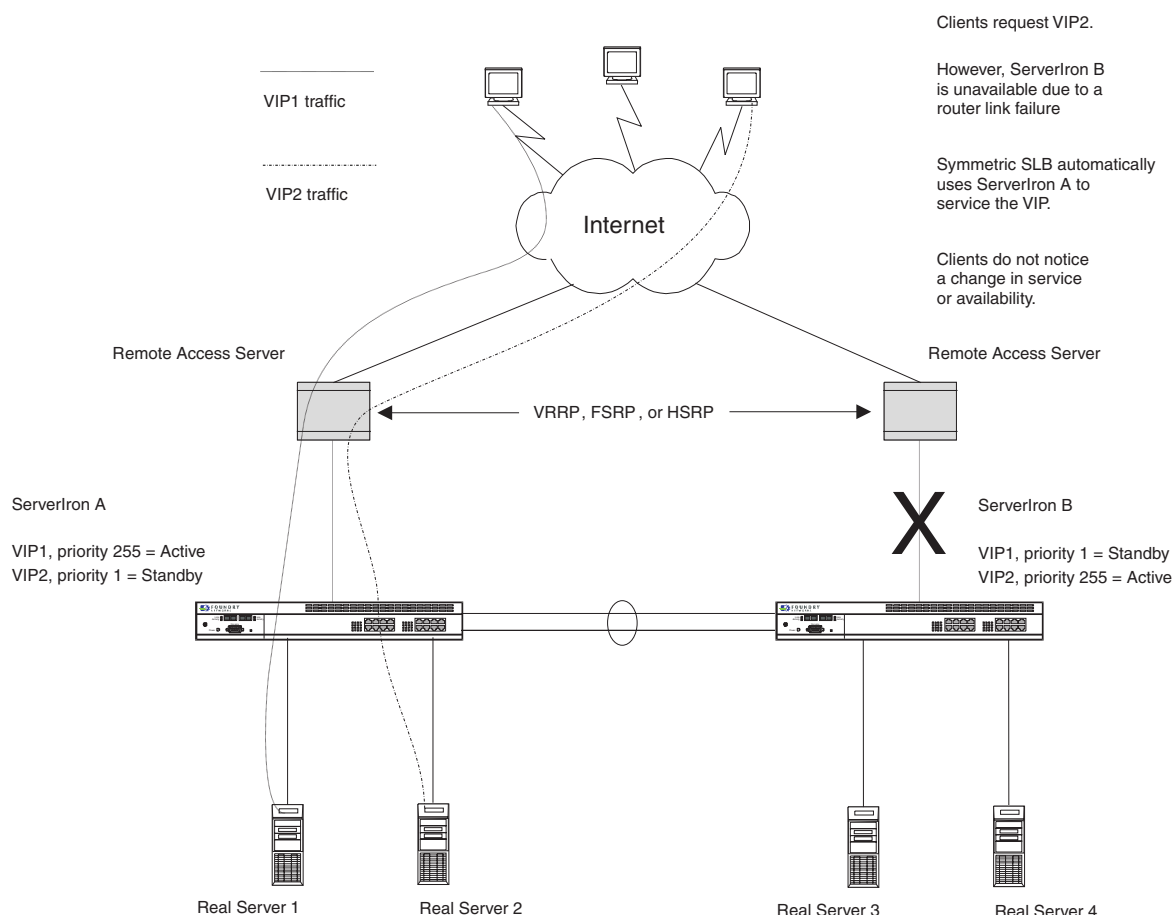
In a Symmetric SLB configuration, each VIP is actively served by only one of the ServerIrons. The other ServerIrons are standbys for that VIP, although they may each simultaneously be the active ServerIron for other VIPs. You determine which ServerIron is the active ServerIron for a VIP by assigning a priority to each VIP. The ServerIron that has the highest priority for a specific VIP is the active ServerIron for the VIP by default. The other ServerIrons have lower priorities for the VIP and are standbys for that VIP. Only if the ServerIron that has the highest priority for the VIP becomes unavailable does another ServerIron (with the next highest priority for the VIP) assume service for the VIP.

To configure Symmetric SLB, you configure the same VIPs on multiple ServerIrons that have Layer 2 connectivity, then assign a different SLB priority to each VIP on each of the ServerIrons. For example, to configure two ServerIrons for SLB, configure the same VIPs on each of the ServerIrons. On one of the ServerIrons, assign half of the VIPs a priority of 1 (lowest) and assign the other VIPs a priority of 255 (highest). Assign the reverse priorities to the VIPs on the other ServerIron.

A typical Symmetric SLB configuration uses a redundant set of real servers connected to each ServerIron. The VIPs and their contents are identical on each pair of servers. The only difference for each VIP is the priority you assign the VIP on a particular virtual server. You can configure a priority from 1 – 255 and you can use up to 255 ServerIrons in a Symmetric SLB configuration.

Figure 2.10 shows an example of Symmetric SLB.

**Figure 2.10 Symmetric SLB automatically compensates for unavailable equipment and links**



## Link-Level Redundancy

Symmetric SLB includes link-level redundancy to provide fault tolerance against failed links.

If a link from a Serverlron to the real servers fails, Symmetric SLB can use an alternate path through another Serverlron running Symmetric SLB to reach the real servers. Link-level redundancy requires no additional configuration. If the Serverlrons have Layer 2 connectivity and you configure Symmetric SLB priorities for the VIPs, then link-level redundancy is automatically enabled.

See “Configuring Symmetric SLB and SwitchBack” on page 7-1 for a detailed application example.

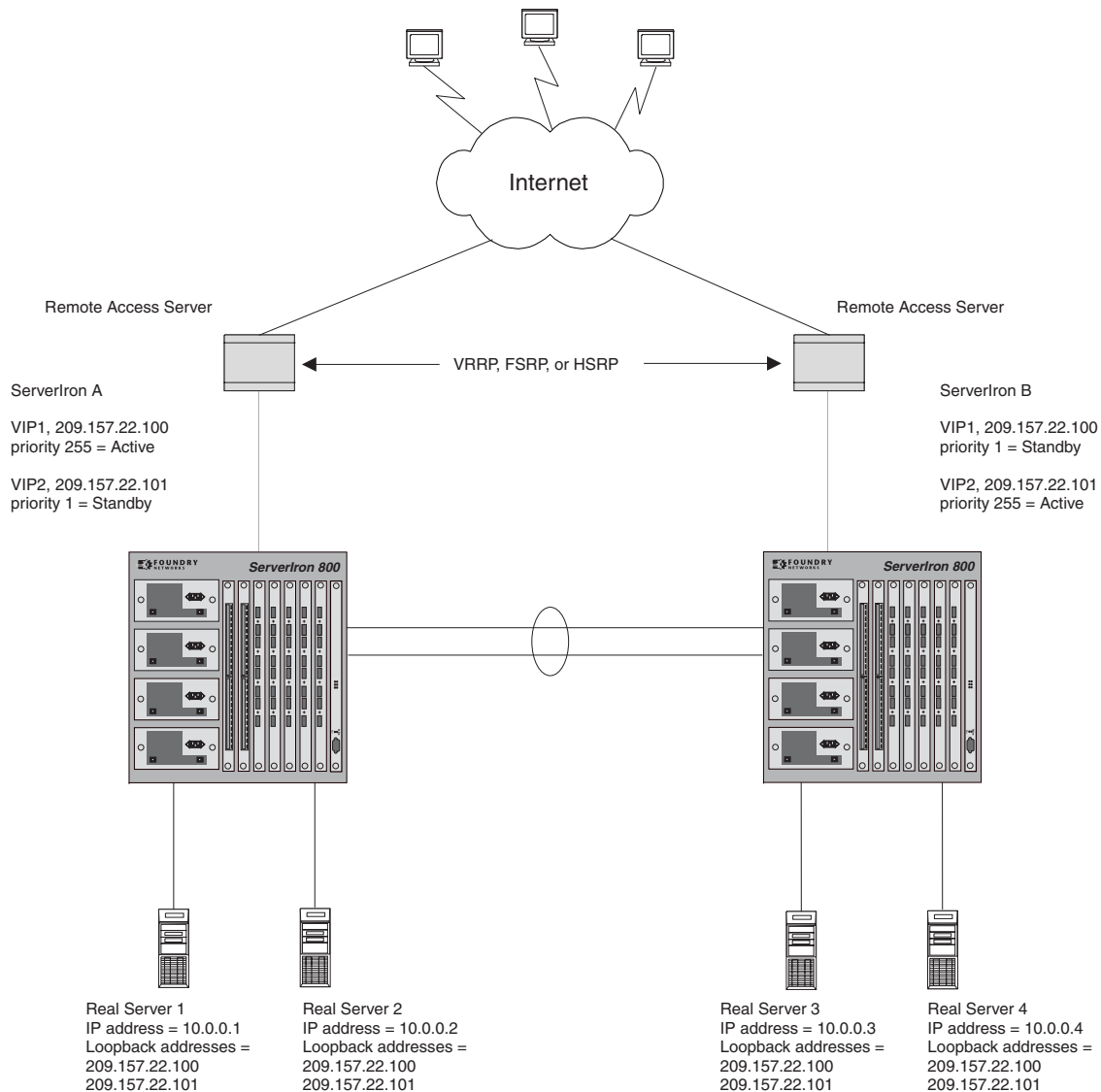
## SwitchBack

SwitchBack configures the Serverlron to instruct real servers to send client responses directly to the clients instead of sending the responses back through the Serverlron. As a result, the clients receive faster response time and the Serverlron is free to support even more sessions to serve more clients. (A connection is two sessions, one in each direction.)

When configured for this feature, the Serverlron sends client requests addressed to a VIP to a load balanced real server, as in standard Server Load Balancing (SLB) configurations. However, to enhance server-to-client response time and to increase the overall connection capacity of the Serverlron, the software in a SwitchBack configuration formats the client request packets in such a way that the real servers respond directly to the clients, instead of sending the responses back through the Serverlron.

Figure 2.11 shows an example of two ServerIron 800s deployed in a Server Load Balancing (SLB) SwitchBack configuration.

**Figure 2.11 ServerIron 800s deployed in SwitchBack configuration**



You configure SwitchBack on an individual TCP/UDP port basis when you configure your virtual servers. The feature requires that you configure a loopback interface on each real server and give the loopback interface the IP address of the VIP. The ServerIron sends the client traffic to the real server without translating the destination address from the VIP into the real server's IP address. Thus, the real server receives the client traffic addressed to its loopback address and responds directly to the client.

The SwitchBack feature can be used on a single ServerIron supporting a single server farm, but is also quite powerful when used on multiple ServerIrons in combination with the Symmetric SLB feature (see "Symmetric Server Load Balancing" on page 2-26).

For configuration and implementation details and a complete implementation example, see "Configuring Symmetric SLB and SwitchBack" on page 7-1.

## Many-To-One TCP/UDP Port Binding

When you associate a VIP with a real server, you make the association for a particular TCP/UDP port. The association of a TCP/UDP port on a VIP with a TCP/UDP port on a real server is called a "port binding". Typical configurations use one VIP-to-real-server binding for a TCP/UDP port. For example, if you bind VIP 192.29.2.2 to real server 10.0.0.1 for port 80 (the well-known HTTP port), generally you do not then create another binding between VIP 192.29.2.2 and real server 10.0.0.1 for the same port.

However, if you want to track statistics for individual applications or domain names mapped to the same port, the ServerIron allows you to configure an alias for the port. You configure a separate alias for each additional VIP. For example, if you are associating three VIPs with the same real server, you define two TCP/UDP port aliases, one for each of the additional VIPs. The ServerIron collects and displays statistics and configuration information individually for each VIP, but sends all traffic to the same TCP/UDP port number on the real server.

See "Many-To-One TCP/UDP Port Binding" on page 6-98 for an example application using this feature.

## HTTP Redirect

The remote server support and NAT support described in previous sections allow you to configure geographically-distributed servers that the ServerIron uses as failovers if the local servers are unavailable. A typical configuration with geographically-distributed servers uses source NAT to cause responses from the remote real server to go back to the ServerIron instead of directly to the client. This traffic pattern matches the standard traffic pattern among the ServerIron, the clients, and the real servers.

However, if the links between a remote server and ServerIron are slow and would introduce unacceptable delays, you can enable HTTP redirect. HTTP redirect configures the ServerIron to send an HTTP redirect message to a client when the ServerIron is sending the client's request to a remote server. The HTTP redirect instructs the client to redirect its TCP connection from the VIP to the real IP address of the remote server. After a successful HTTP redirect, the client and remote server communicate directly, not through the ServerIron.

---

**NOTE:** If a client creates a bookmark when communicating directly with a remote server, the bookmark points to the real IP address of the server instead of the VIP. If the client uses the bookmark later to re-access the server, the client bypasses the VIP.

---

## Transparent VIP and Stateless Application Ports

Transparent VIP allows you to configure a ServerIron to transparently load balance a VIP, without owning the VIP address. Use this feature when you want to configure multiple ServerIrons to load balance the same VIP. For example, if you have globally distributed clients that access the same VIP, you can place ServerIrons close to those clients for optimal service, and use the ServerIron to load balance requests for the VIP to locally distributed server farms.

Depending on the network topology, you might also want to configure the application ports on the transparent VIP to be stateless. A stateless port does not use session table entries and the ServerIron does not expect the server reply for the port to come back through the ServerIron. Standard Layer 4 and Layer 7 keepalive health checking relies on session table entries, but you can configure stateless health checking for the stateless ports.

For more information about these features, see "Configuring Symmetric SLB and SwitchBack" on page 7-1.

## Web Switching Features

Web switching allows the ServerIron to make forwarding decisions about HTTP traffic using information in a URL, cookie, or SSL session ID. The ServerIron can perform the following kinds of web switching:

- URL switching directs HTTP requests to a server group using information in the text of a URL string.
- Cookie switching directs HTTP requests to a server group based on information embedded in a cookie in the HTTP header.
- HTTP header hashing internally maps certain kinds of information in an HTTP header to a real server and directs all HTTP requests that contain this information to this real server.

- SSL Session ID switching connects a client to the same server to which it had previously established an SSL (Secure Sockets Layer) connection.

## URL Switching

URL switching is the ServerIron's ability to direct HTTP requests to a server, or group of servers, using information in the text of a URL string. The ServerIron examines the contents of a URL string and makes a decision about where to send the packet based on selection criteria in user-defined policies. If text in the URL string matches the selection criteria, the HTTP request is sent to a load-balanced server group specified in the policy.

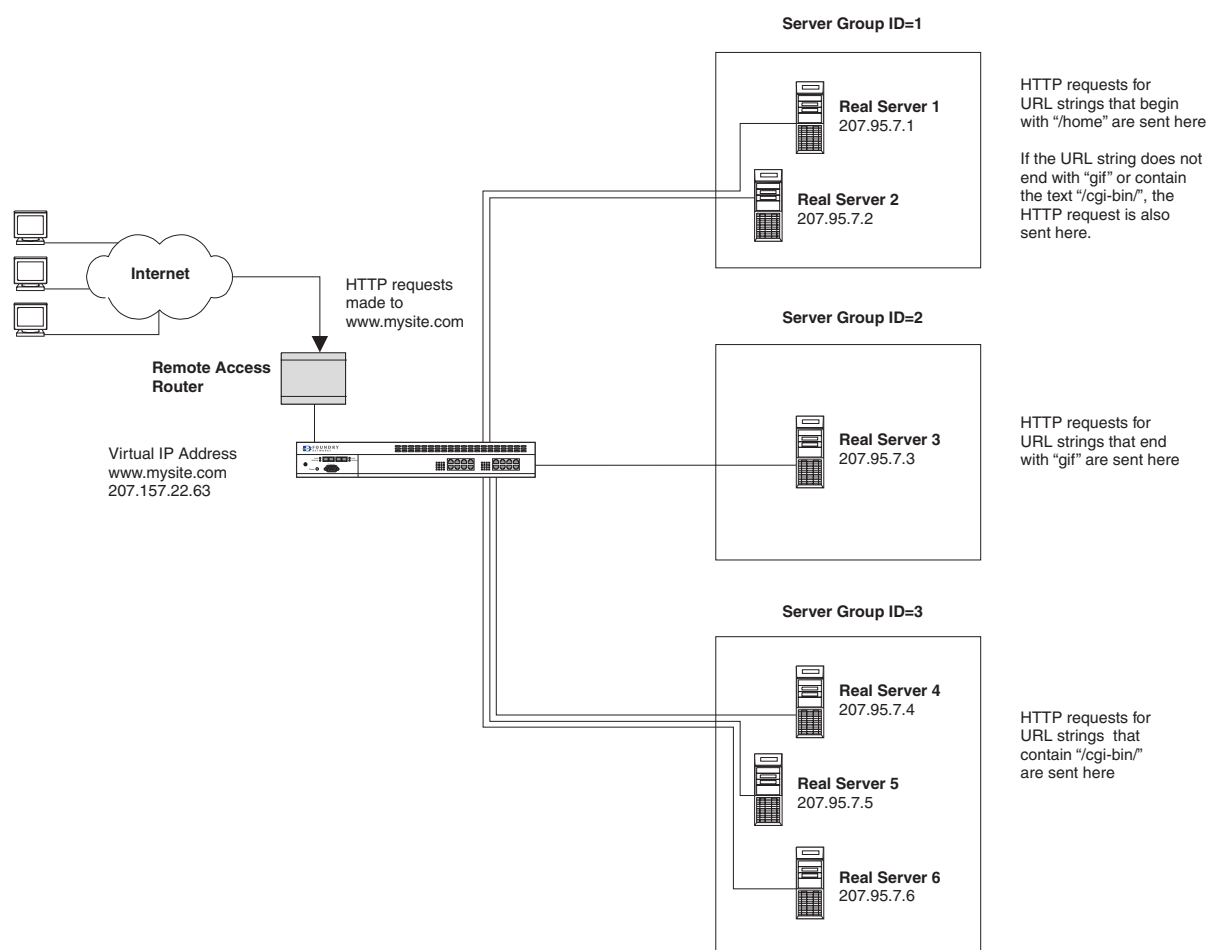
Unlike standard server load balancing, which requires that the same content be on all load-balanced real servers, URL switching allows you to place different web content on different servers. For example, you can place image files on one group of servers and CGI applications on another group. Information in the URL string determines to which server group HTTP requests are sent.

The diagram in Figure 2.12 illustrates a basic example of URL switching. The ServerIron is connected to three groups of load-balanced real servers. The server group with ID = 1 contains the /home directory for the web site. The server group with ID = 2 contains all the GIF files for the web site. The server group with ID = 3 contains all the CGI applications for the web site.

The ServerIron has URL switching policies in place that cause HTTP requests to be directed as follows:

- HTTP requests containing URL strings that start with the text "/home" are sent to server group ID = 1.
- HTTP requests containing URL strings that end with the text ".gif" are sent to server group ID = 2.
- HTTP requests containing URL strings that have the text "/cgi-bin/" anywhere within are sent to server group ID = 3.
- If a URL string does not start with the text "/home", end with the text ".gif", or contain the text "/cgi-bin/", the HTTP request is sent to server group ID = 1.

Figure 2.12 Example of a URL switching configuration



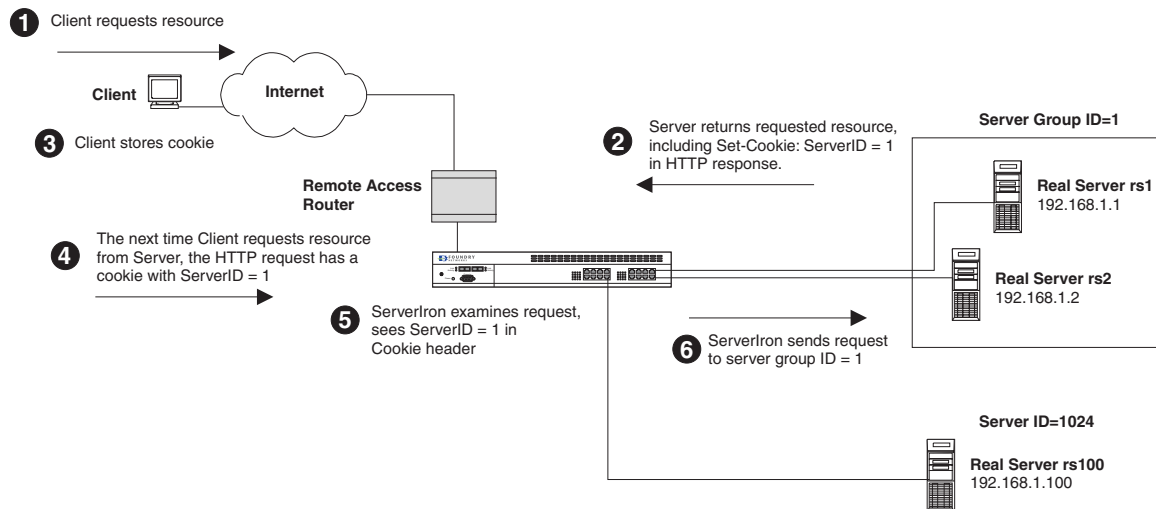
For configuration and implementation details, see "Configuring URL Switching" on page 11-1.

## Cookie Switching

Cookie switching is the ServerIron's ability to direct HTTP requests to a server group based on information embedded in a cookie in the HTTP header. You configure your server to send a cookie when responding to a request from a client. The client stores the cookie, and the next time the client requests information from the server, the cookie specifies which server group should handle the request. In this way, you can ensure that requests from a particular client are always handled by a particular server group, even across sessions.

Figure 2.13 illustrates how cookie switching works.

**Figure 2.13 How cookie switching works**



For configuration and implementation details, see “Configuring Cookie Switching” on page 11-22.

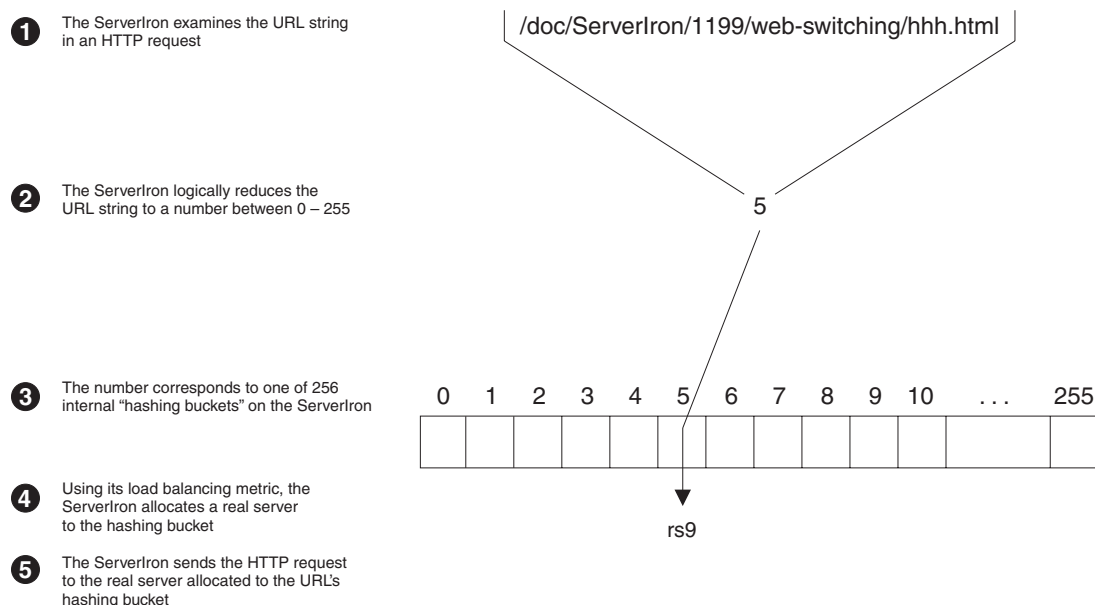
## HTTP Header Hashing

In HTTP header hashing, the ServerIron examines information in the HTTP request (either the Cookie header or the URL string) and internally maps this information to one of the real servers bound to the virtual server. This HTTP request and all future HTTP requests that contain this information then always go to the same real server.

For example, an HTTP request might have a URL string that consists of the text “/download/files/myfile.html”. The ServerIron would internally map this URL string to a real server bound to the virtual interface. The next time an HTTP request that has this exact same URL string comes into the VIP, it would go to the same real server as the first one did.

Figure 2.14 illustrates how the ServerIron uses HTTP header hashing to direct HTTP requests to a real server.



**Figure 2.14 Using HTTP header hashing to select a real server**

The ServerIron can also perform HTTP header hashing using the Cookie header or a segment of the URL string. For configuration and implementation details, see "Configuring HTTP Header Hashing" on page 11-30.

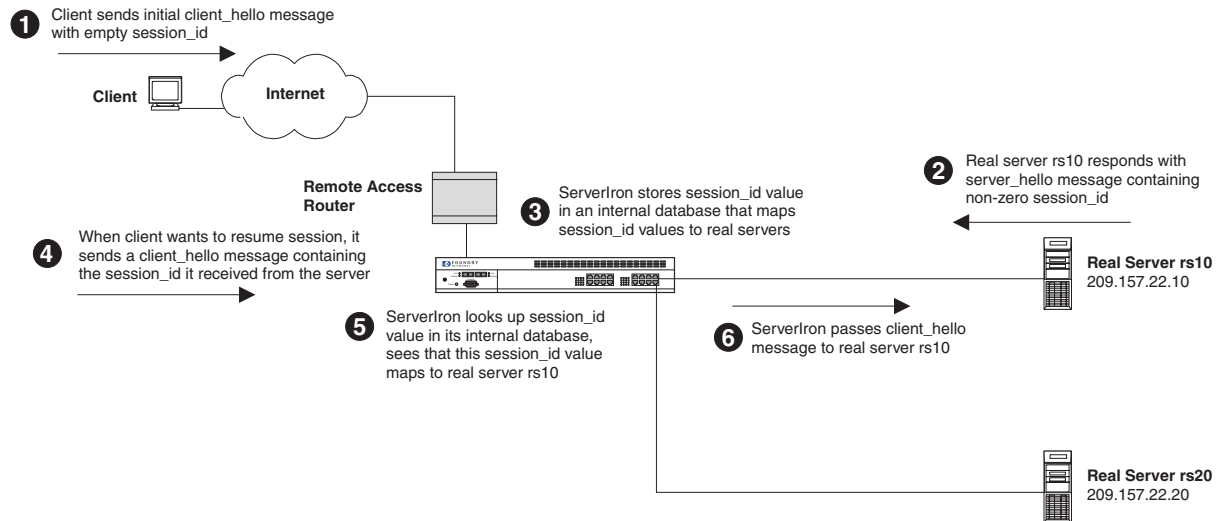
## SSL Session ID Switching

SSL (Secure Sockets Layer) is a protocol for secure World Wide Web connections. The SSL protocol protects your confidential information with server authentication, data encryption and message integrity. SSL is layered beneath application protocols such as HTTP, Telnet, FTP, Gopher, and NNTP, and layered above the connection protocol TCP/IP. This allows SSL to operate independently of the Internet application protocols. With SSL implemented on both the client and server, your Internet communications are transmitted in encrypted form, ensuring privacy.

In order for SSL to work, all the SSL connections between a client and server must reach the same host. SSL connections come in sequentially on particular ports; only one is open at a time. However, each must go to the same server. *SSL Session ID switching* is the ServerIron's ability to connect a client to the same server to which it had previously established an SSL connection.

Figure 2.15 illustrates how SSL Session ID switching works.

**Figure 2.15 How the ServerIron uses an SSL Session ID to select a real server**



For configuration and implementation details, see “Configuring SSL Session ID Switching” on page 11-40.

## Transparent Cache Switching (TCS) Features

TCS allows a ServerIron or Foundry backbone switch to detect and switch web traffic to a local cache server within the network. A single ServerIron (or hot standby pair) can provide transparent cache switching for up to 1024 web cache servers.

Cache servers process web queries faster and more efficiently by temporarily storing details about repetitive web queries locally, reducing the number of external inquiries required to process a web query. By limiting the number of queries sent to remote web servers, the overall WAN access capacity required is lessened as is the overall operating cost for WAN access.

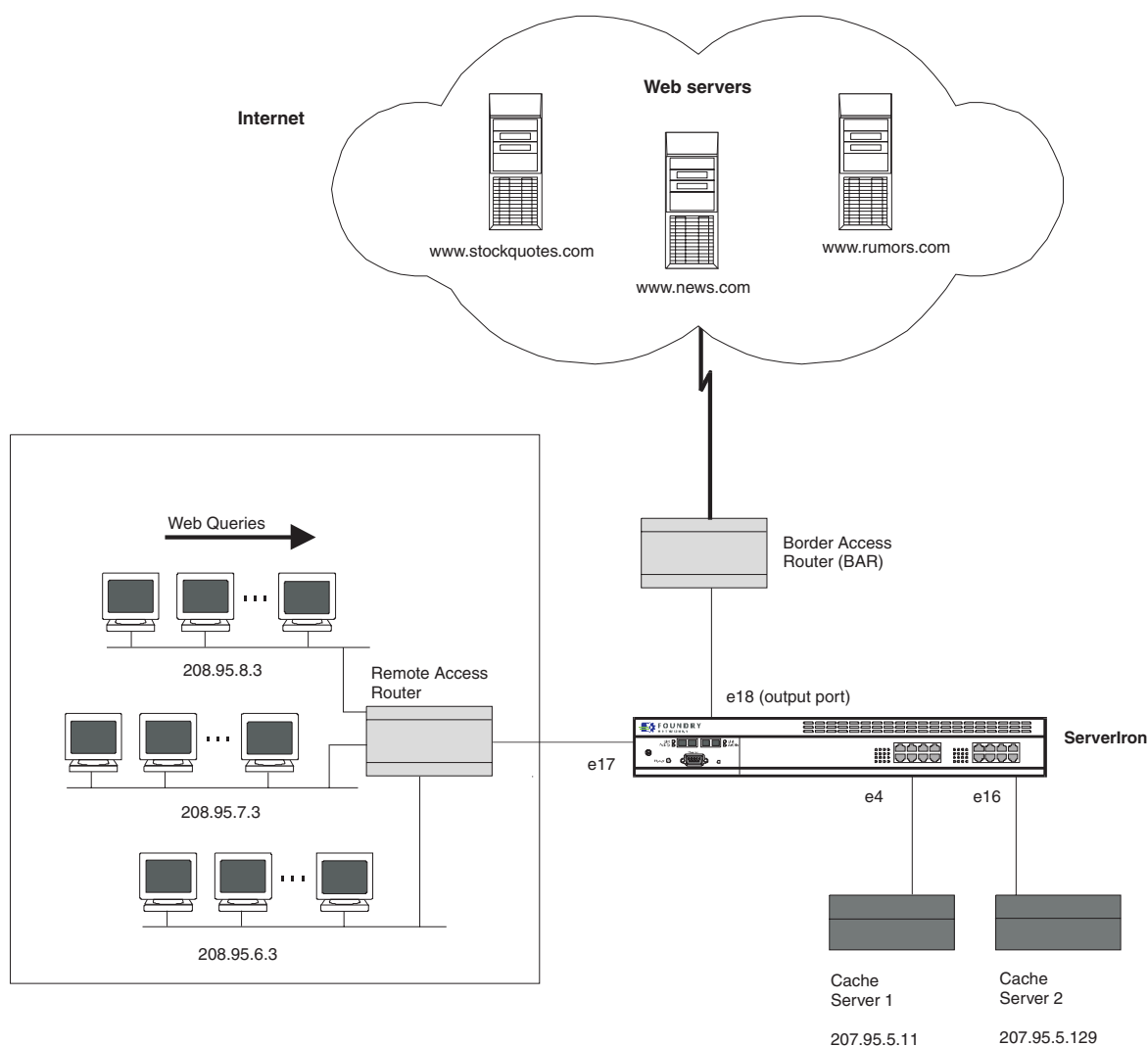
Foundry switches increase the reliability of transparent caching within a network by supporting redundant web cache server configurations known as web cache server groups, as well as supporting redundant paths to those server groups with the **server backup** option.

### How TCS Works

A Foundry ServerIron or backbone switch operating as a transparent cache switch detects and forwards all web (HTTP) traffic to an available cache server. The cache server then processes the query and forward the response back to the user through the attached Foundry switch, as shown in Figure 2.16.

The cache server determines how the web query will be handled by pulling from its local information stores and facilitating that information with external web queries on the WAN, as needed, to complete the query.

The Foundry switch provides the detection and switching of those HTTP packets forwarded from the cache server. This process is known as “transparent” cache switching because it is transparent to the end user. The end user continues to see the web site page(s) as expected in answer to his or her query and is unaware that the access point to the information is through the cache server. Additionally, because TCS works with the default settings of web browsers, no configuration changes are required on the client station, which further adds to the transparency of the feature.

**Figure 2.16 Logical representation of transparent caching**

### Response to Cache Server Failures

Web cache servers are grouped with other cache servers to provide for automatic recovery from a failed or otherwise out-of-service web cache server. The ServerIron monitors the availability of the cache servers in the group. If a web cache server failure occurs, the switch detects the failure and directs subsequent requests to the next available cache server or forwards the request directly to the WAN link. You can gain further reliability by using redundant ServerIrons, thereby eliminating any single point of failure in the server group network path.

### Stateful Caching

Stateful caching provides the following services:

- Minimization of route flap problems.
- Graceful shutdown of transparent cache servers.
- Ability to set maximum connections allowed for a cache server.
- Use of filters to control caching based on source and destination addresses.
- Advanced statistics for TCS.

By default, stateful TCS is enabled on a switch when TCS is active (enabled).

## Minimization of Route Flap Problems

When a route change causes web query traffic to be moved from an non-cached path to a cached path, no TCS is performed on the active connections.

---

**NOTE:** When the opposite transition occurs—web query traffic moving from a cached to non-cached path—the ServerIron takes no action because the traffic is no longer visible to the ServerIron.

---

## Configurable Maximum Connections for Cache Server

You can set the maximum number of connections that a cache server will accept. By setting a limit, you can avoid a condition where the capacity threshold of a cache server is exceeded.

When a server reaches the maximum defined connection threshold, an SNMP trap is sent. When all the cache servers in a cache group reach the maximum connection threshold, the ServerIron sends the client requests to the Internet.

## Advanced Statistics

TCS provides the following advanced statistics:

- Current connections on a per cache basis
- Peak connections on a per cache basis
- Total connections on a per cache basis
- Packet counts to and from cache on a per-cache basis
- Octet counts to and from cache on a per-cache basis

## Cache Route Optimization (CRO)

Typically a ServerIron sits between a border access router (BAR) and a remote access server (RAS) where the BAR connects to the Internet/Intranet. The RAS forwards the client HTTP traffic to the ServerIron, which re-directs the traffic to the cache servers. When a border router is configured as the default router for the cache servers, all traffic sent towards the browsing clients behind the RAS must first go to the BAR.

At Layer 3, the cache server sends its response to the IP address of the client (or to the ServerIron if source NAT is enabled on the ServerIron). However, at Layer 2 the cache server sends its response to the MAC address of its default gateway. In configurations where the default gateway is the BAR, this traffic pattern can cause significant (and unnecessary) BAR performance degradation and poor response time as perceived by the clients.

The Cache Route Optimization (CRO) feature sends traffic from a cache server toward the RAS. When you enable the feature, the ServerIron uses information in its Layer 4 session table and the ServerIron's traffic pattern recognition capabilities to redirect the traffic directly toward the clients instead of sending the traffic needlessly to the BAR.

CRO is disabled by default. See "Cache Route Optimization (CRO)" on page 10-35 for more information about the feature.

## Policy-Based Cache Failover (CFO)

In some TCS configurations, the ServerIron is connected to the clients and also to the Internet through the same router. Moreover, in some cases the router contains a policy to forward HTTP requests to a virtual IP address if the packet containing the request matches a filter configured on the router.

In this case, the ServerIron dutifully forwards the request to a cache server. However, if the requested address is not cached on the server, the cache server tries to find the requested site in the Internet by sending the request back to the ServerIron, which drops the packet.

You can configure the ServerIron to send the request back to the router for forwarding to the Internet by adding the virtual IP address to the cache group. By adding the virtual IP address, you enable the CFO feature.

For more information and a detailed example, see "Policy-Based Cache Failover (CFO)" on page 10-41.

## Firewall Load Balancing

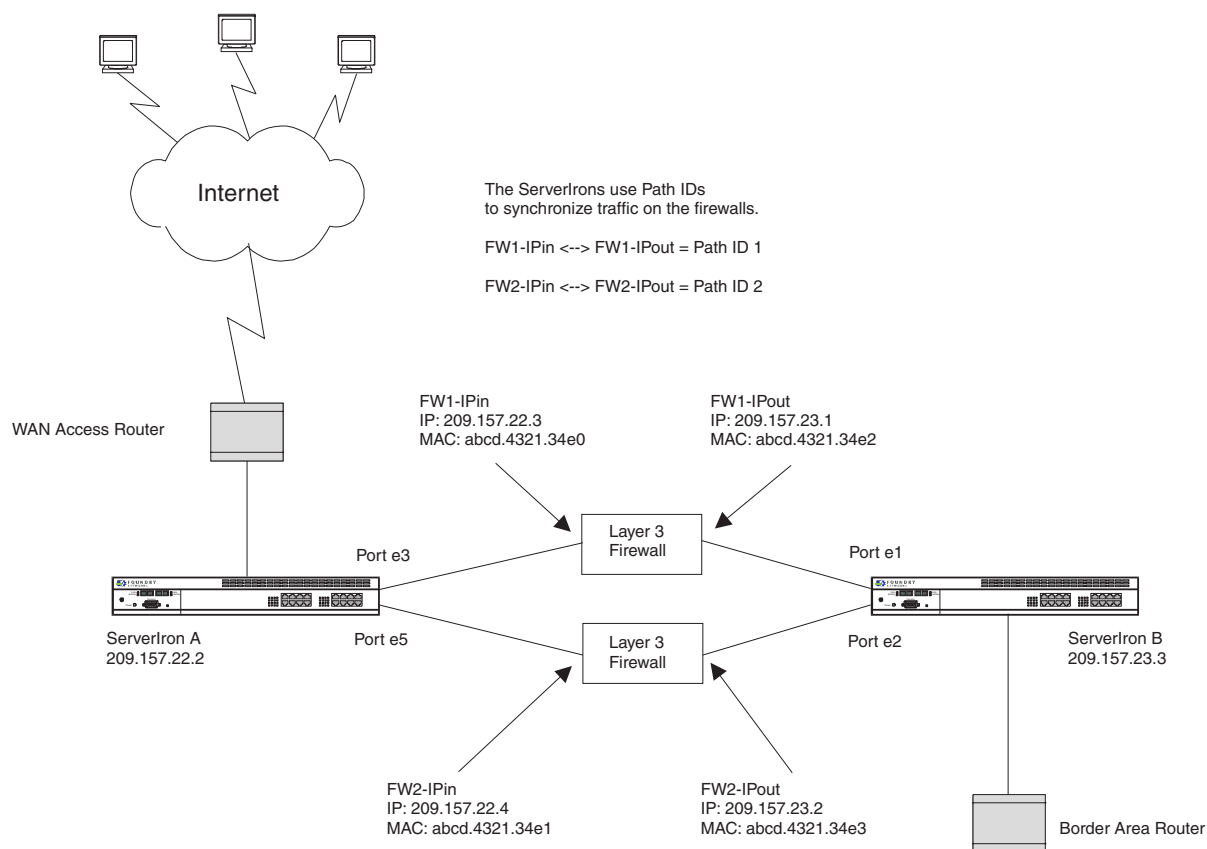
Firewall load balancing enhances overall firewall performance by distributing traffic across multiple firewalls and synchronizing the connections to eliminate unnecessary reauthentications.

To implement basic firewall load balancing, you configure two ServerIrons, one on each side of your firewalls.

- One of the ServerIrons is on the Internet side of the firewalls.
- The other ServerIron is on the private network side.

This configuration is sometimes called a “sandwich” firewall load balancing configuration. Figure 2.17 shows an example.

**Figure 2.17 Sandwich firewall configuration**



When you configure the ServerIrons, one of the parameters you configure on each ServerIron is a path for each firewall. A path associates a specific ServerIron port with a path ID. In the case of Layer 3 firewalls, a path also identifies the next-hop interface (typically the firewall interface).

Paths enable the two ServerIrons to use the same firewall for a connection between a host inside the firewall and a host outside the firewall. By ensuring that the same firewall is used for both ends of the connection, the ServerIrons reduce the overhead that can be caused by revalidations of the same connection source or destination by different firewalls.

You can load balance for up to 8 firewalls. The ServerIron uses the same hash distribution mechanism as the one used for TCS to balance traffic across the firewalls within a group.

In addition, for asynchronous firewalls (firewalls that do not exchange or coordinate information about client traffic among themselves), the ServerIron uses the paths that you configure to synchronize the firewall traffic. By synchronizing the traffic for the firewalls, the ServerIron reduces the number of authentications that the firewalls must perform.

See “Configuring Server Load Balancing” on page 6-1 for more information.

## IronClad Firewall Load Balancing

For added reliability, you can configure pairs of ServerIrons on each side of the firewalls. One of the ServerIrons in each pair is active and performs the firewall load balancing. The other ServerIron remains in standby mode but takes over if the active ServerIron becomes unavailable.

See the *Foundry ServerIron Firewall Load Balancing Guide* for information.

## IP Forwarding

IP forwarding enables you to configure IP interfaces on the ServerIron and use the interfaces as default gateways for your real servers and other devices.

IP forwarding provides an alternative to using source IP addresses to multinet the ServerIron. Source IP addresses allow the ServerIron to be in multiple sub-nets but the ServerIron remains a Layer 2 and Layer 4 – 7 switch. When IP forwarding is disabled, the ServerIron does not route IP traffic.

When you enable IP forwarding on the ServerIron, the ServerIron becomes a Layer 3 IP router in addition to a Layer 2 and Layer 4 – 7 switch. You can configure multiple IP interfaces in different sub-nets to multinet the ServerIron. The ServerIron uses an IP route table to select paths for forwarding traffic.

IP forwarding also allows hosts to establish management sessions with the ServerIron even when the hosts and the ServerIron’s management IP address are in different sub-nets.

See “Configuring IP Forwarding” on page 13-1.

## Network Address Translation (NAT)

You can configure the ServerIron to perform standard **Network Address Translation (NAT)**. NAT enables private IP networks that use nonregistered IP addresses to connect to the Internet. Configure NAT on the Foundry device at the border of an inside network and an outside network (such as the Internet). NAT translates the internal local addresses to globally unique IP addresses before sending packets to the outside network. NAT also allows a more graceful renumbering strategy for organizations that are changing service providers or voluntarily renumbering into Classless Interdomain Routing (CIDR) blocks.

See “Network Address Translation (NAT)” on page 2-38.

---

**NOTE:** Standard NAT support is separate from the virtual IP address features provided for Server Load Balancing (SLB). For example, standard NAT is not related to source IP addresses used for multinetting the ServerIron, performing health checks on remote servers, and so on.

---

## Layer 2 Switching Features

The following sections describe the Layer 2 switching features listed in Table 2.4 on page 2-5. For more information about these features and how to configure them, see the *Foundry Switch and Router Installation and Basic Configuration Guide*.

### MAC Switching

All Foundry devices support MAC switching. **MAC switching** enables intelligent wire-speed bridging of Layer 2 packets. The first time a Foundry device receives a packet from a given MAC destination, the device makes an entry in its Layer 2 cache. The entry consist of the packet’s source MAC address and the port on which the device received the packet.

When the device receives a bridge packet destined for the cached address, the device does not need to send the packet as a broadcast through all the ports within the broadcast domain. Instead, the device can intelligently send the packet only through the port to which the destination device is connected. Thus, even though Layer 2 domains

are typically broadcast domains, MAC switching enhances performance in the domain by reducing the amount of broadcast traffic in the domain.

In addition, Foundry routers that are enabled for MAC switching can switch traffic for route protocols that are not supported in the routing software. If IPX routing is disabled on a router, the router can switch the IPX packets instead.

To avoid accumulating stale cache entries, Foundry devices use an aging mechanism. The aging mechanism removes a learned entry from the cache after the entry has remained unused for a specified interval (by default, 300 seconds). You can change or disable the aging interval.

See the “Configuring Basic Features” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

By default, all ports in a Foundry device belong to a common Layer 2 broadcast domain, VLAN 1. You can configure port-based VLANs (Virtual LANs) to create smaller broadcast domains that use subsets of the device’s ports. See the “Configuring Virtual LANs (VLANs)” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## Static MAC Entries

MAC entries that the Foundry device learns and caches are subject to an aging time. After a cached entry remains unused for the duration of the aging time, the software removes the entry from the Layer 2 cache. If you want certain MAC addresses to always be present in the device’s Layer 2 address table, you can add them as static entries.

A **static MAC entry**, like a cached (dynamic) MAC entry, maps a MAC address to the Foundry device’s port attached to that device.

Unlike cached MAC entries, static MAC entries provide the following benefits:

- You can assign a QoS priority to a static MAC entry.
- You can specify VLAN membership for a static MAC entry.
- A static entry prevents broadcast storms that can be caused when a server’s MAC entry is removed. For example, if a server goes down long enough for the server’s entry to age out, the Foundry device sends packets addressed to the server as broadcasts until the device relearns the cache entry for the server.

You can specify port priority (QoS) and VLAN membership (VLAN ID) for the MAC address. On switches, you also can specify the device type (router or host) for the entry.

---

**NOTE:** On Foundry routers, you also can create static IP routes, ARP entries, and RARP entries. The ServerIron and other Foundry switches support only static MAC addresses.

---

See the “Configuring Basic Features” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## Standard Spanning Tree Protocol (STP)

The **Spanning Tree Protocol** (STP) is a protocol for detecting and eliminating logical loops in a Layer 2 broadcast domain. STP is described in the IEEE 802.1d bridge protocols standard. STP is supported on all Foundry switches and routers.

STP also ensures that the device uses the most efficient path when multiple paths exist between ports. Moreover, if a selected path fails, STP searches for and then establishes an alternate path to prevent or limit retransmission of data.

STP is disabled by default on routers but is enabled by default on the ServerIron and other switches.

See the “Configuring Basic Features” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## IronSpan STP Enhancements

IronSpan is a set of Layer 2 features that extend the operation of standard STP. IronSpan enables you to fine tune standard STP and avoid some of its limitations. IronSpan includes the following features:

- **Fast Port Span** – By default, devices running Fast Port Span perform Spanning Tree Protocol (STP) convergence in four seconds instead of 30 or more seconds for certain ports connected to end stations.
- **Fast Uplink Span** – Enhances STP by allowing a Foundry device with redundant uplinks to quickly resume forwarding, in just four seconds. This feature is similar to Fast Port Span but applies to certain inter-switch links on Foundry devices, instead of Foundry links to end stations.

For more information and configuration procedures, see the “IronSpan Spanning Tree Protocol (STP)” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## Trunk Groups

A **trunk group** is a set of ports that provide a high speed link between two Foundry devices or between a Foundry device and a server. A trunk group can consist of up to four ServerIron physical ports and provides the bandwidth of those ports combined. Thus, a trunk group containing four 1 Gbps ports can provide up to four Gbps of bi-directional traffic.

In addition to enabling load sharing of traffic, trunk groups provide redundant, alternate paths for traffic. Thus, if a link in a trunk group fails, the device still uses the other links in the trunk group.

A ServerIron trunk group can consist of two or four ports. You can configure up to four trunk groups on a ServerIron.

See the “Configuring Basic Features” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## Port-Based Virtual LANs (VLANs)

By default, all ports in a Foundry device belong to a common Layer 2 broadcast domain. When the device sends a broadcast packet, the packet goes out all active ports. A **port-based VLAN** (Virtual LAN) is a subset of ports on a Foundry device that constitutes a Layer 2 broadcast domain.

Port-based VLANs can reduce the likelihood and severity of broadcast storms by reducing the number of ports affected by a storm. In addition, for devices such as servers that can cause broadcast storms, you can add static MAC entries for the devices and assign the static entries to a VLAN.

Each port-based VLAN maintains a separate spanning tree. (See “Standard Spanning Tree Protocol (STP)” on page 2-39.)

See the “Configuring Virtual LANs (VLANs)” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## VLAN Tagging

Foundry switches support 802.1q VLAN tagging. **VLAN tagging** is a method of identifying a packet as a member of a VLAN. VLAN tagging enables you to configure ports on multiple switches into a single VLAN. Using tagged VLANs can ease network management and ensures interoperability with other devices.

When a switch sends a packet that is a member of a tagged VLAN, the switch “tags” the packet to indicate its VLAN membership. Other switches that support VLAN tagging recognize the tag and process the packet according to its VLAN membership.

See the “Configuring Virtual LANs (VLANs)” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## MAC Filters

A **MAC filter** enables you to explicitly permit or deny switching of a Layer 2 packet received by the Foundry device. When the device receives a Layer 2 packet for switching, the device checks the packet’s contents against the defined MAC filters. If the packet matches a filter, the system takes the action specified in the filter.



- If the action is permit, the system allows the packet to be switched.
- If the action is deny, the system immediately drops the packet.

To ensure security, if a packet does not match any of the MAC filters defined on the system, the system drops the packet by default. To configure the system to permit packets by default, you must define the last MAC filter in the filter list to allow all packets.

MAC filters can evaluate packets based on criteria such as source address and mask, destination address and mask, and protocol type (IP, ARP, and so on).

See the “Configuring Basic Features” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

---

**NOTE:** You cannot use Layer 2 filters to filter Layer 4 information. To filter Layer 4 information, use IP filters. See “IP Filters” on page 2-16.

---

---

**NOTE:** MAC filters are not supported on the FastIron Workgroup Switch.

---

## Address-Lock Filters

An **address-lock filter** restricts the number of MAC addresses that a switch can learn from a specific port. After the switch learns the specified number of MAC addresses from the port, the switch stops learning addresses received on that port. In addition, the switch does not accept or forward traffic on the port unless the traffic contains one of the source or destination MAC addresses locked for the port.

Address-lock filters apply only to Layer 2 traffic and do not affect Layer 3 or Layer 4 traffic on the locked ports.

Unlike addresses learned from other ports, addresses learned from a locked port are not subject to aging.

See the “Configuring Basic Features” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## Dynamic Host Configuration Protocol (DHCP) Assist

**DHCP Assist** allows a Foundry switch to assist a router that is performing multinetting on its interfaces as part of its DHCP relay function. DHCP eliminates the need to manually assign IP addresses to clients. Instead of each client having a statically configured IP address, clients petition a server for IP addresses when the clients are booted.

DHCP Assist ensures that a DHCP server that manages multiple IP sub-nets can readily recognize the requester’s IP sub-net, even when that server is not on the client’s local LAN segment. The Foundry switch does this by stamping the correct gateway IP address into a DHCP discovery packet on behalf of the router.

---

**NOTE:** DHCP assist applies only to the ServerIron and other Foundry switches. To configure a Foundry router to assist DHCP packets, use the UDP Helper feature. See the “Configuring Basic Features” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

---

For configuration information about DHCP Assist and UDP Helper, see the “Configuring Basic Features” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## IP Multicast Containment

**IP multicast containment** allows Foundry switches to limit switching of IP multicast packets to only those ports on the switch that are identified as IP multicast members. Foundry switches can provide IP multicast containment in either of the following modes:

- **Passive** – The switch listens for Internet Group Membership Protocol (IGMP) packets and forwards them to the appropriate ports.
- **Active** – The switch actively sends out host queries to identify IP multicast groups on the network and inserts this information into the IGMP packets.

Routers in the network generally handle host queries. Unless your configuration does not contain a router to provide this service, use IP multicast containment in the passive mode.

---

# Chapter 3

## Installing the ServerIron

This chapter describes how to install Foundry ServerIrons and attach them to your network.

### Unpacking a System

The Foundry ServerIron ships with all the following items. Please review the list below and verify the contents. If any items are missing, contact the place of purchase.

#### Package Contents

- Foundry ServerIron
- 115V AC power cable
- Rack mount brackets and mounting screws
- CD-ROM containing software and user documentation (including this guide)
- Warranty card

#### General Requirements

To manage the system, you need the following items for serial connection to the ServerIron:

- A management station, such as a PC running a terminal emulation application.
- A straight-through EIA/TIA DB-9 serial cable (F/F). The serial cable can be ordered separately from Foundry Networks (part number CC). If you prefer to build your own cable, see the pinout information in “Attaching a PC or Terminal” on page 3-3.

You use the serial connection to perform basic configuration tasks including assigning a management IP address and network mask to the system. This information is required for managing the system using the Web management interface or IronView or using the CLI through Telnet.

### Summary of Installation Procedures

Follow the steps below to install your ServerIron. Details for each step are provided later in this chapter.

1. Ensure that the physical environment that will host the device has the proper cabling and ventilation. See “Preparing the Installation Site” on page 3-3.
2. Verify that the system and module LEDs are registering the proper LED state after power-on of the system. See “Verifying Proper Operation” on page 3-3.

3. A terminal or PC serial port connection is all that is required to support configuration on the ServerIron. See “Attaching a PC or Terminal” on page 3-3.
4. Before attaching equipment to the device, you need to assign an interface IP address for the sub-net on which the device will be located. Initial IP address assignment is done using the CLI with a direct serial connection. Subsequent IP address assignments can be performed using the Web management interface. See “Assigning IP Addresses” on page 3-5.

---

**NOTE:** The ServerIron allows you to configure up to four additional IP sub-net addresses. Depending on the network topology, you might need to configure additional addresses. See “Source IP Address” on page 6-30.

---

5. Foundry devices can be installed on a desktop or in an equipment rack. See “Mounting the ServerIron in an Equipment Rack” on page 3-6.
6. Once the device is physically installed, plug the device into a nearby power source that adheres to the regulatory requirements outlined in this guide. See “Powering On the ServerIron” on page 3-7.
7. Once you power on the device and assign IP addresses, the system is ready to accept network equipment. See “Connecting Network Devices” on page 3-7.
8. Test IP connectivity to other devices by pinging them. See “Testing Connectivity” on page 3-10.
9. Configure a system name, contact, and location. See “Configuring a System Name, Contact, and Location” on page 3-10.
10. Continue configuring the device using the CLI or the Web management interface. See “Managing the ServerIron” on page 3-11.
11. Secure access to the device. See *Foundry Security Guide*.

## Installation Precautions

Follow these precautions when installing a Foundry device:

---

**WARNING:** Make sure the rack or cabinet housing the device is adequately secured to prevent it from becoming unstable or falling over.

---

---

**WARNING:** Mount the devices you install in a rack or cabinet as low as possible, placing the heaviest device at the bottom and progressively placing lighter devices above.

---

### CAUTION:

- Make sure that the power source circuits are properly grounded, then use the power cord supplied with the device to connect it to the power source.  
  
If the installation requires a different power cord than the one supplied with the device, make sure you use a power cord displaying the mark of the safety agency that defines the regulations for power cords in your country. The mark is your assurance that the power cord can be used safely with the device.
  - Ensure that the device does not overload the power circuits, wiring, and over-current protection. To determine the possibility of overloading the supply circuits, add the ampere ratings of all devices installed on the same circuit as the device. Compare this total with the rating limit for the circuit. The maximum ampere ratings are usually printed on the devices near the AC power connectors.
  - Do not install the device in an environment where the operating ambient temperature might exceed 40° C (104° F).
  - Make sure the air flow around the front, sides, and back of the device is not restricted.
-

---

**NOTE:** If you are installing a chassis device, see the “Installing a Foundry Layer 2 Switch or Layer 3 Switch” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide* for additional cautions and installation instructions.

---

## Preparing the Installation Site

### Cabling Infrastructure

Ensure that the proper cabling is installed in the site. See “Connecting Network Devices” on page 3-7 for a summary of supported cabling types and their specifications.

### Installation Location

Before installing the device, plan its location and orientation relative to other devices and equipment. Allow at least 3" of space at the front of the device for the twisted-pair, fiber-optic, and power cabling. Also, allow a minimum of 3" of space between the sides and the back of the device and walls or other obstructions.

## Verifying Proper Operation

Before mounting the device in its network location, verify that the device is working properly by plugging it into a power source and verifying that it passes its self test.

If your device has more than one power supply installed, repeat this procedure for each power supply.

1. Connect the power cord supplied with the ServerIron to the power connector on the power supply at the rear of the device.
2. Insert the other end into a properly grounded electrical outlet.

---

**NOTE:** The devices do not have power switches. They power on when you connect a power cord to the device and to a power source.

If your installation requires a different power cord than that supplied with the device, make sure you obtain a power cord displaying the mark of the safety agency that defines the regulations for power cords in your country. The mark is your assurance that the power cord can be used safely with the device.

---

3. Check the LEDs on the front of the device. When the system is first powered on, the lights simultaneously light as the system goes through system and module diagnostics. After the initial diagnostics are complete, none of the port LEDs light up until a cable is connected. For more details on specific LED conditions after system start-up, see “LEDs” on page 2-3.

## Attaching a PC or Terminal

To assign an IP address, you must have access to the **Command Line Interface (CLI)**. The CLI is a text-based interface that can be accessed through a direct serial connection to the device and through Telnet connections. The CLI is described in detail in the *Foundry ServerIron Command Line Interface Reference*.

Foundry devices do not have an IP address when shipped from the factory. You need to assign an IP address using the CLI. You can access the CLI by attaching a serial cable to the Console port. After you assign an IP address, you also can access the system through Telnet and through the Web management interface.

### To attach a management station using the serial port:

1. Connect a PC or terminal to the serial port of the system using a straight-through cable. The serial port has a male DB-9 connector.

---

**NOTE:** You need to run a terminal emulation program on the PC.

---

2. Open the terminal emulation program and set the session parameters as follows:

- Baud: 9600 bps
- Data bits: 8
- Parity: None
- Stop bits: 1
- Flow control: None

When you establish the serial connection to the system, press Enter to display the following CLI prompt in the terminal emulation window: `ServerIron>`

If you see this prompt, you are now connected to the system and can proceed to “Assigning IP Addresses” on page 3-5.

**NOTE:** If you are configuring a Foundry switch that you have upgraded with ServerIron code, the command prompt might be different.

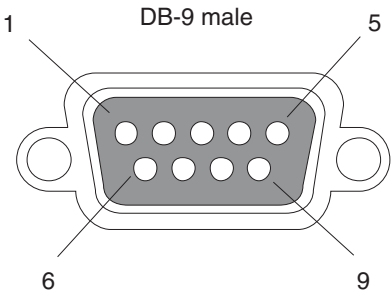
You can customize the prompt by changing the system name. See “Configuring a System Name, Contact, and Location” on page 3-10.

If you do not see one of these prompts:

1. Make sure the cable is securely connected to your PC and to the Foundry system.
2. Check the settings in your terminal emulation program. In addition to the session settings listed above, make sure the terminal emulation session is running on the same serial port you attached to the Foundry system.

The EIA/TIA 232 serial communication port serves as a connection point for management by a PC or SNMP workstation. Foundry switches and Layer 3 Switches come with a standard male DB-9 connector, shown in Figure 3.1.

**Figure 3.1 Serial port pin and signalling details**

Pin Assignment	Pin Number	Switch Signal
	1	Reserved
	2	TXD (output)
	3	RXD (input)
	4	Reserved
	5	GND
	6	Reserved
	7	CTS (input)
	8	RTS (output)
	9	Reserved

Most PC serial ports also require a cable with a female DB-9 connector.

Terminal connections will vary, requiring either a DB-9 or DB-25 connector, male or female.

Serial cable options between a Foundry switch or router and a PC or terminal are shown in Figure 3.2.

**NOTE:** As indicated in Figure 3.1 and Figure 3.2, some of the wires should not be connected. If you do connect the wires that are labeled “Not Used” or “Not Connected”, you might get unexpected results with some terminals.

**Figure 3.2 Serial port pin assignment showing cable connection options to a terminal or PC**

DB-9 to DB-9 Female Switch			Terminal or PC			DB-9 to DB-25 Female Switch			Terminal or PC		
1	Reserved		1			1	Reserved		8		
2		►	2			2		►	3		
3	◄		3			3	◄		2		
4	Reserved		4			4	Reserved		20		
5			5			5			7		
6	Reserved		6			6	Reserved		6		
7	◄		7			7	◄		4		
8		►	8			8		►	5		
9	Reserved		9			9	Reserved		22		

## Assigning IP Addresses

Foundry devices are not pre-configured at the factory with IP addresses. You must assign an IP address using the serial connection to the CLI before you can manage the system using the other management interfaces.

Foundry devices support both classical IP network masks (Class A, B, and C sub-net masks, and so on) and prefix masks.

- To enter a classical network mask, enter the mask in IP address format. For example, enter "209.157.22.99 255.255.255.0" for an IP address with a Class-C sub-net mask.
- To enter a network mask using prefix addressing, enter a forward slash ( / ) and the number of bits in the mask immediately after the IP address. For example, enter "209.157.22.99/24" for an IP address that has a network mask with 24 significant ("mask") bits.

By default, the CLI displays network masks in classical IP address format (example: 255.255.255.0). You can change the display to prefix-addressing format. See the "Configuring IP and IP/RIP" chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

**NOTE:** If your network uses a BootStrap Protocol (BootP) server or a Dynamic Host Configuration Protocol (DHCP) server, you can allow the Foundry device to obtain IP information from the server.

### To assign a management IP Address to a ServerIron:

- At the opening CLI prompt, enter **enable**.

```
ServerIron> enable
```

- Enter the following command at the Privileged EXEC level prompt (for example, `ServerIron#`), then press Enter. This command erases the factory test configuration if still present:

```
ServerIron# erase startup-config
```

**WARNING:** Use this step only for new systems. If you enter this command on a system you have already configured, the command erases the configuration. If you accidentally do erase the configuration on a configured system, enter the **write memory** command to save the running configuration to the startup-config file.

- Access the configuration level of the CLI by entering the following command:

```
ServerIron# configure terminal
ServerIron(config)#
```

4. Set the IP and mask addresses.

```
ServerIron(config)# ip address 192.22.3.44 255.255.255.0
```

5. Set a default gateway address for the switch.

```
ServerIron(config)# ip default-gateway 192.22.3.1
```

**Syntax:** enable [<password>]

**Syntax:** configure terminal

**Syntax:** ip address <ip-addr> <ip-mask>

or

**Syntax:** ip address <ip-addr>/<mask-bits>

**Syntax:** ip default-gateway <ip-addr>

---

**NOTE:** If you plan to deploy the ServerIron in a multinetted environment, you can add up to four additional source IP addresses. The source IP addresses allow the ServerIron and the real servers, remote access server (RAS), and border access router (BAR) to be on different sub-nets. See “Source IP Address” on page 6-30.

---

## Mounting the ServerIron in an Equipment Rack

You can install the ServerIron on a desktop or in an equipment rack.

---

**WARNING:** Make sure the rack or cabinet housing the ServerIron is adequately secured to prevent it from becoming unstable or falling over.

---

---

**WARNING:** Mount the devices you install in a rack or cabinet as low as possible, placing the heaviest device at the bottom and progressively placing lighter devices above.

---

---

**NOTE:** If you are installing a ServerIron 400 or ServerIron 800, see the “Installing a Foundry Layer 2 Switch or Layer 3 Switch” chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide* for installation instructions for chassis devices.

---

### Desktop Installation

1. Set the ServerIron on a flat desktop, table, or shelf. Make sure that adequate ventilation is provided for the system—a 3-inch clearance is recommended on each side.
2. Go to “Testing Connectivity” on page 3-10.

### Rack Mount Installation

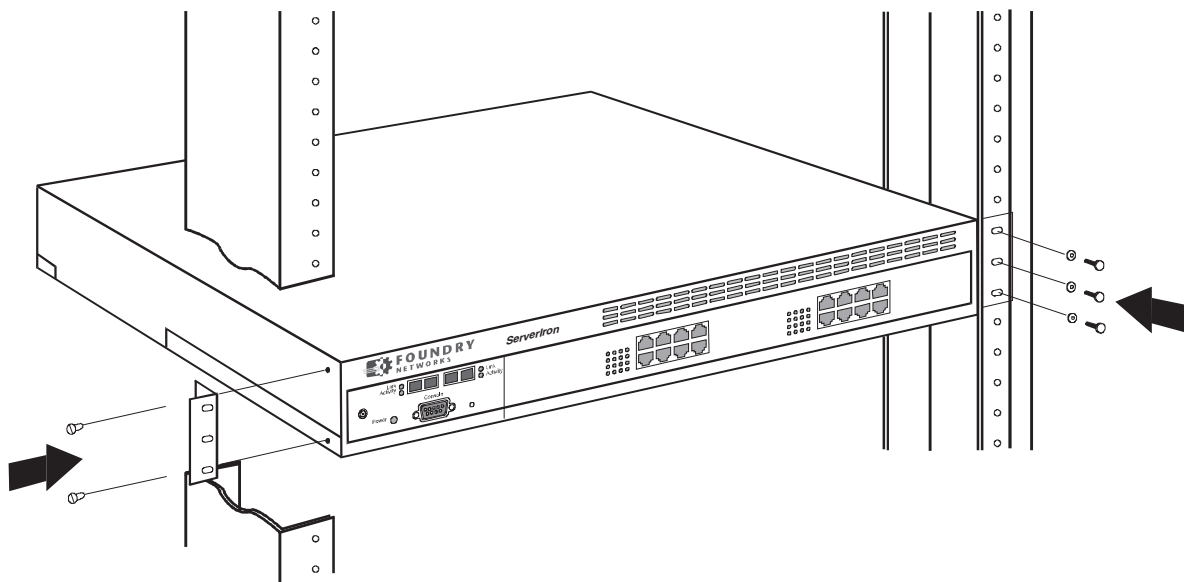
---

**NOTE:** You need a #2 Phillips-head screwdriver for installation.

---

1. Remove the rack mount kit from the shipping carton. The kit contains two L-shaped mounting brackets and mounting screws.
2. Attach the mounting brackets to the sides of the ServerIron as illustrated in Figure 3.3.
3. Attach the ServerIron in the rack as illustrated in Figure 3.3.
4. Proceed to “Testing Connectivity” on page 3-10.



**Figure 3.3** Installing a ServerIron in an equipment rack

## Powering On the ServerIron

After you complete the physical installation of the ServerIron, you can power on the system.

1. Remove the power cord from the shipping package.
2. Attach the AC power cable to the AC connector on the rear panel.
3. Insert the power cable plug into a 115V/120V outlet.

---

**NOTE:** Foundry devices are designed to provide uninterrupted service even when you insert or remove modules. Therefore, the systems do not have separate on/off power switches. To turn the system off, simply unplug the power cord(s).

---



---

**NOTE:** The socket should be installed near the equipment and should be easily accessible.

---



---

**NOTE:** If the outlet is not rated 115/120V, stop and get the appropriate cable for the outlet.

---

## Connecting Network Devices

Foundry devices can support connections to other vendors' routers, switches, and hubs as well other Foundry switches and routers.

---

**NOTE:** You can configure trunk groups to connect the ServerIron to other devices. A trunk group is a set of physical ports that are logically configured as a single port. If you configure a trunk group, you must use the "server" parameter instead of the default "switch" parameter, regardless of the device type you are connecting to. This applies even if the other device is a ServerIron. See the *Foundry ServerIron Command Line Interface Reference* for the CLI command syntax.

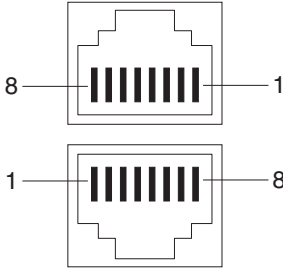
---

The 8-port, 16-port, and 24-port ServerIrons all use 10BaseT/100BaseTX ports with RJ-45 jacks for standard unshielded twisted pair (UTP/Category 5) cable connections.

If you have the optional gigabit uplink or are upgrading a Foundry FastIron Backbone switch or a stackable Turbolron/4 switch (not the Turbolron/8), these devices support the following types of interfaces:

- 10BaseT/100BaseTX ports come with RJ45 jacks for standard unshielded twisted pair (UTP/Category 5) cable connections.
- 100BaseFX ports come equipped with SC(MMF) connectors.
- 1000BaseSX ports come equipped with SC(MMF) connectors.
- 1000BaseLX ports come equipped with SC(SMF) connectors.
- 1000BaseLH ports come equipped with SC(SMF) connectors.

**Figure 3.4 Pin assignment and signalling for 10BaseTX and 100BaseTX ports**

Pin Assignment	10BaseT Pin Number	MDI-X ports	100BaseTX and 1000BaseT Pin Number	MDI-X ports
	1	RD+	1	RD+
	2	RD-	2	RD-
	3	TD	3	TD
	4	Not used	4	CMT
	5	Not used	5	CMT
	6	TD-	6	TD-
	7	Not used	7	CMT
	8	Not used	8	CMT

## Cable Length

**Table 2.1 Cable length summary table**

	Fiber Type	Core Diameter (microns)	Modal Bandwidth (MHz*km)	Minimum Range (meters)
1000Base-SX	MMF	62.5	160	2 – 200 <sup>a</sup>
	MMF	62.5	200	2 – 275 <sup>b</sup>
	MMF	50	400	2 – 500
	MMF	50	500	2 – 550 <sup>c</sup>
1000Base-LX	MMF	62.5	500	2 – 550
	MMF	50	400	2 – 550
	MMF	50	500	2 – 550
	SMF	9	n/a	2 – 5000

- The TIA 568 building wiring standard specifies 160/500 MHz\*km MMF (Multi-mode Fiber).
- The international ISO/IEC 11801 building wiring standard specifies 200/500 MHz\*km MMF.
- The ANSI Fibre Channel specification specifies 500/500 MHz\*km 50 micron MMF and 500/500 MHz\*km fiber has been proposed for addition to ISO/IEC 11801.

- 100BaseTX: Cable length should not exceed 100 meters in keeping with the IEEE 802.3 standard.

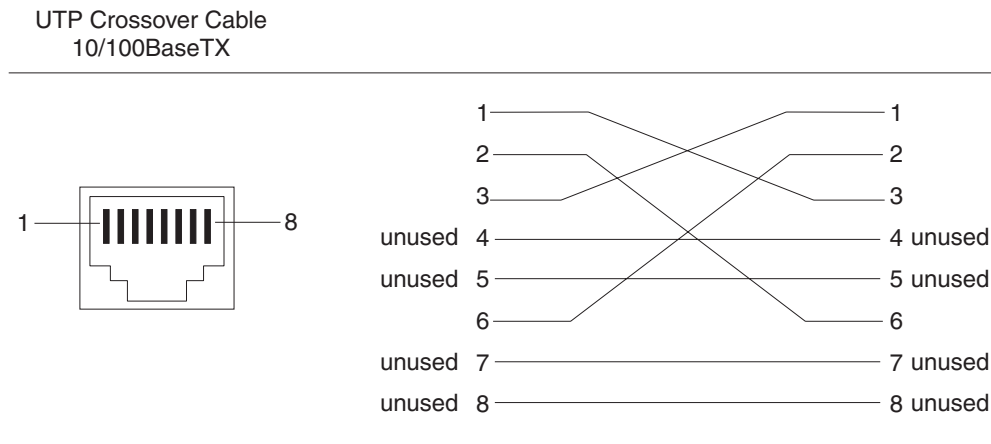
- 100BaseFX: Cable length should not exceed 2 kilometers.
- 1000BaseSX: Cable length should not exceed 550 meters when operating with multi-mode cabling.
- 1000BaseLX:
  - Cable length of 2 – 550 meters is supported on 62.5  $\mu\text{m}$  multi-mode fiber (MMF) cabling.
  - Cable length of 2 – 550 meters is supported on 50  $\mu\text{m}$  multi-mode fiber (MMF) cabling.
  - Cable length of 2 – 5000 meters is supported on 9  $\mu\text{m}$  single-mode fiber (SMF) cabling.
- 1000BaseLH: Cable length should not exceed 70 kilometers.

**NOTE:** Cable installation and network configuration will affect overall transmission capability. The numbers provided above represent the accepted recommendations of the various standards. For network-specific recommendations, consult your local Foundry reseller system engineer.

## Connecting to Ethernet or Fast Ethernet Hubs

For UTP connections to Ethernet or Fast Ethernet hubs, a 10/100 Mbps switch, or another switch or router, a crossover cable is required (Figure 3.5). If the hub is equipped with an uplink port, it will require a straight-through cable rather than a crossover cable.

**Figure 3.5 UTP crossover cable**



## Connecting to Workstations, Servers, or Routers

Straight-through UTP cabling is required for direct UTP attachment to workstations, servers, or routers using network interface cards (NICs).

Fiber cabling with SC connectors is required for direct attachment to Gigabit NICs or switches and routers.

## Troubleshooting Network Connections

- For the indicated port, verify that both ends of the cabling (at the ServerIron and at the connected device) are snug.
- Verify that the ServerIron and the connected device are both powered on and operating correctly.
- Verify that you have used the correct cable type for the connection:
  - For twisted-pair connections to an end node, use straight-through cabling.
  - For fiber-optic connections, verify that the transmit port on the ServerIron is connected to the receive port on the connected device, and that the receive port on the ServerIron is connected to the transmit port on the connected device.

- Verify that the port has not been disabled through a configuration change. You can use the CLI. If you have configured an IP address on the device, you also can use the Web management interface.
- If the procedures above do not resolve the problem, try using a different port or a different cable.

## Testing Connectivity

After you install the network cables, you can test network connectivity to other devices by pinging those devices. You also can perform trace routes.

### Pinging an IP Address

To verify that a ServerIron can reach another device through the network, enter a command such as the following at any level of the CLI:

```
ServerIron> ping 192.33.4.7
```

**Syntax:** ping <ip-addr> | <hostname> [count <num>] [timeout <msec>] [ttl <num>] [size <byte>] [no-fragment] [quiet] [verify] [data <1 – 4 byte hex>] [brief]

See the *Foundry ServerIron Command Line Interface Reference* for information about the parameters.

### Tracing a Route

To determine the path through which a Foundry device can reach another device, enter a command such as the following at any level of the CLI on the Foundry device:

```
ServerIron> traceroute 192.33.4.7
```

**Syntax:** traceroute <host-ip-addr> [minttl <value>] [maxttl <value>] [timeout <value>] [numeric]

See the *Foundry ServerIron Command Line Interface Reference* for information about the command syntax.

## Configuring a System Name, Contact, and Location

You can configure a system name, contact, and location for a Foundry switch or router and save the information locally in the configuration file for future reference. This information is not required for system operation but is recommended. When you configure a system name, the name replaces the default system name in the CLI command prompt. For example, if the system is a ServerIronXL, the system name you configure replaces "ServerIron" in the command prompt.

The name, contact, and location each can be up to 32 alphanumeric characters.

Here is an example of how to configure a name, system contact, and location using the CLI:

```
ServerIron(config)# hostname Oakland
Oakland(config)# snmp-server contact Jack London
Oakland(config)# snmp-server location oakcabldg519
Oakland(config)# end
Oakland# write memory
```

**Syntax:** hostname <text>

**Syntax:** snmp-server contact <text>

**Syntax:** snmp-server location <text>

## Managing the ServerIron

You can manage a Foundry device using the following interfaces:

- Command Line Interface (CLI) – a text-based interface accessible through a direct serial connection or a Telnet session.
- Web management interface – A GUI-based management interface accessible through an HTTP (web browser) connection.

### Logging on Through the CLI

After you assign an IP address to the ServerIron or switch, you can access the CLI either through a direct serial connection to the device or through a local or remote Telnet session.

You can initiate a local Telnet or SNMP connection by attaching a straight-through RJ-45 cable to a port and specifying the assigned management station IP address.

### Logging On Through the Web Management Interface

To use the Web management interface, open a web browser and enter the IP address of the Foundry device in the browser's Location or Address field. When the Login dialog is displayed, enter a valid user name and password for read-only or read-write access.

By default, you can use the user name "get" and the default read-only password "public" for read-only access. However, for read-write access, you must enter "set" for the user name, and enter the read-write community string configured on the ServerIron for the password. Beginning with software release 05.0.00, there is no default read-write community string. You must add one using the CLI. See "Establishing SNMP Community Strings" on page 3-11.

As an alternative to using the SNMP community strings to log in, you can configure the ServerIron to secure Web management access using local user accounts or Access Control Lists (ACLs). See *Foundry Security Guide*.

## Establishing SNMP Community Strings

The default passwords for Web management access are actually the SNMP community strings configured on the device.

- The default read-only community string is "public". To open a read-only Web management session, enter "get" and "public" for the user name and password.
- Beginning with software release 05.1.00, there is no default read-write community string. Thus, by default, you cannot open a read-write management session using the Web management interface. You first must configure a read-write community string using the CLI. Then you can log on using "set" as the user name and the read-write community string you configure as the password.

You can configure as many additional read-only and read-write community strings as you need. The number of strings you can configure depends on the memory on the device. There is no practical limit.

The Web management interface supports only one read-write session at a time. When a read-write session is open on the Web management interface, subsequent sessions are read-only, even if the session login is "set" with a valid read-write password.

---

**NOTE:** If you delete the startup-config file, the device automatically re-adds the default "public" read-only community string the next time you load the software.

---

---

**NOTE:** As an alternative to the SNMP community strings, you can secure Web management access using local user accounts or ACLs. See the *Foundry Security Guide*.

---

## Encryption of SNMP Community Strings

The software automatically encrypts SNMP community strings. Users with read-only access or who do not have access to management functions in the CLI cannot display the strings. For users with read-write access, the strings are encrypted in the CLI but are shown in the clear in the Web management interface.

Encryption is enabled by default. You can disable encryption for individual strings or trap receivers if desired. See the next section for information about encrypting the strings.

## Adding an SNMP Community String

To add a community string, use either of the following methods. When you add a community string, you can specify whether the string is encrypted or clear. By default, the string is encrypted.

### USING THE CLI

To add an encrypted community string, enter commands such as the following:

```
ServerIron(config)# snmp-server community private rw
ServerIron(config)# write memory
```

**Syntax:** [no] snmp-server community [0 | 1] <string> ro | rw

The **0 | 1** parameter specifies whether you want the software to encrypt the string (**1**) or show the string in the clear (**0**). The default is **1**.

The <string> parameter specifies the community string name. The string can be up to 32 characters long.

The **ro | rw** parameter specifies whether the string is read-only (**ro**) or read-write (**rw**).

The command in the example above adds the read-write SNMP community string “private”. When you save the new community string to the startup-config file (using the **write memory** command), the software adds the following command to the file:

```
snmp-server community 1 <encrypted-string> rw
```

To add a non-encrypted community string, you must explicitly specify that you do not want the software to encrypt the string. Here is an example:

```
ServerIron(config)# snmp-server community 0 private rw
ServerIron(config)# write memory
```

The command in this example adds the string “private” in the clear, which means the string is displayed in the clear. When you save the new community string to the startup-config file, the software adds the following command to the file:

```
snmp-server community 0 private rw
```

## Displaying the SNMP Community Strings

To display the configured community strings, enter the following command at any CLI level:

```
ServerIron(config)# show snmp server
```

**Syntax:** show snmp server

See the *Foundry Switch and Router Command Line Interface Reference* for an example of the information displayed by the command.

---

**NOTE:** If display of the strings is encrypted, the strings are not displayed. Encryption is enabled by default.

---

### USING THE WEB MANAGEMENT INTERFACE

---

**NOTE:** To make configuration changes, including changes involving SNMP community strings, you must first configure a read-write community string using the CLI. Alternatively, you must configure another authentication method and log on to the CLI using a valid password for that method.

---

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.

**NOTE:** If you have configured the device to secure Web management access using local user accounts, you must instead enter the user name and password of one of the user accounts. See the *Foundry Security Guide*.

**Management**

Web Management:	<input type="radio"/> Disable	<input checked="" type="radio"/> Enable
SNMP:	<input type="radio"/> Disable	<input checked="" type="radio"/> Enable
TELNET:	<input type="radio"/> Disable	<input checked="" type="radio"/> Enable
Telnet Authentication:	<input checked="" type="radio"/> Disable	<input type="radio"/> Enable
Telnet Time Out:	<input type="text" value="0"/>	
Telnet Password:	<input type="text"/>	

[\[Web Preference\]](#)
[\[User Account\]](#)
[\[Authentication Methods\]](#)
[\[System Log\]](#)  
[\[Community String\]](#)
[\[Trap\]](#)
[\[Trap Receiver\]](#)

[\[Home\]](#)
[\[Site Map\]](#)
[\[Logout\]](#)
[\[Save\]](#)
[\[Frame Enable\]](#)
[\[Disable\]](#)
[\[TELNET\]](#)

2. Select the [Management](#) link from the System configuration panel to display the following panel.
3. Select the [Community String](#) link to display the SNMP Community String panel, as shown in the following example. This example shows the table listed for a system that is configured only with the default read-only community string "public".

**SNMP Community String**

Type	Community String	
get	public	<input type="button" value="Delete"/>
Type	Community String	

[\[Add Community String\]](#)

[\[Home\]](#)
[\[Site Map\]](#)
[\[Logout\]](#)
[\[Save\]](#)
[\[Frame Enable\]](#)
[\[Disable\]](#)
[\[TELNET\]](#)

4. Select the [Add Community String](#) link to display a panel such as the following.

**SNMP Community String**

Type:	<input type="radio"/> Get	<input checked="" type="radio"/> Set
Community String:	<input type="text" value="private"/>	
Encrypt:	<input checked="" type="checkbox"/>	

[\[Show\]](#)

[\[Home\]](#)
[\[Site Map\]](#)
[\[Logout\]](#)
[\[Save\]](#)
[\[Frame Enable\]](#)
[\[Disable\]](#)
[\[TELNET\]](#)

5. Select the community string type:

- Select Get for a read-only string.
  - Select Set for a read-write string.
6. Enter the community string in the Community String field.
  7. Select the Encrypt checkbox to remove the checkmark if you want to disable encryption of the string display. Encryption prevents other users from seeing the string in the CLI or Web management interface. If you disable encryption, other users can view the community string. Encryption is enabled by default.  
  
To re-enable encryption, select the checkbox to place a checkmark in the box.
  8. Click the Add button to save the change to the device's running-config file.
  9. Repeat steps 5 – 7 for each string you want to add. You can add as many strings as you need. The limit depends only on the available system memory.
  10. Select the Save link at the bottom of the dialog. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.



---

## Chapter 4

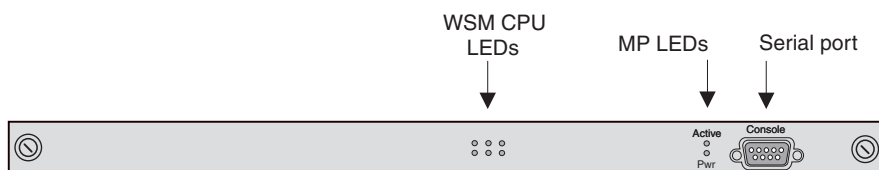
# Using the Web Switching Management Module

This chapter describes the Foundry ServerIron Web Switching Management Module and how to configure and manage it.

The Web Switching Management Module provides the intelligence for Foundry's ServerIron 400 and ServerIron 800, chassis-based ServerIrons.

Figure 4.1 shows the Web Switching Management Module.

**Figure 4.1 Web Switching Management Module**



As shown in Figure 4.1, the Web Switching Management Module does not have network interfaces but does have a serial management interface. In addition, the module has status LEDs for the Management Processor (MP) and the web switching CPUs (WSM CPUs), described below. See “Status LEDs” on page 4-27.

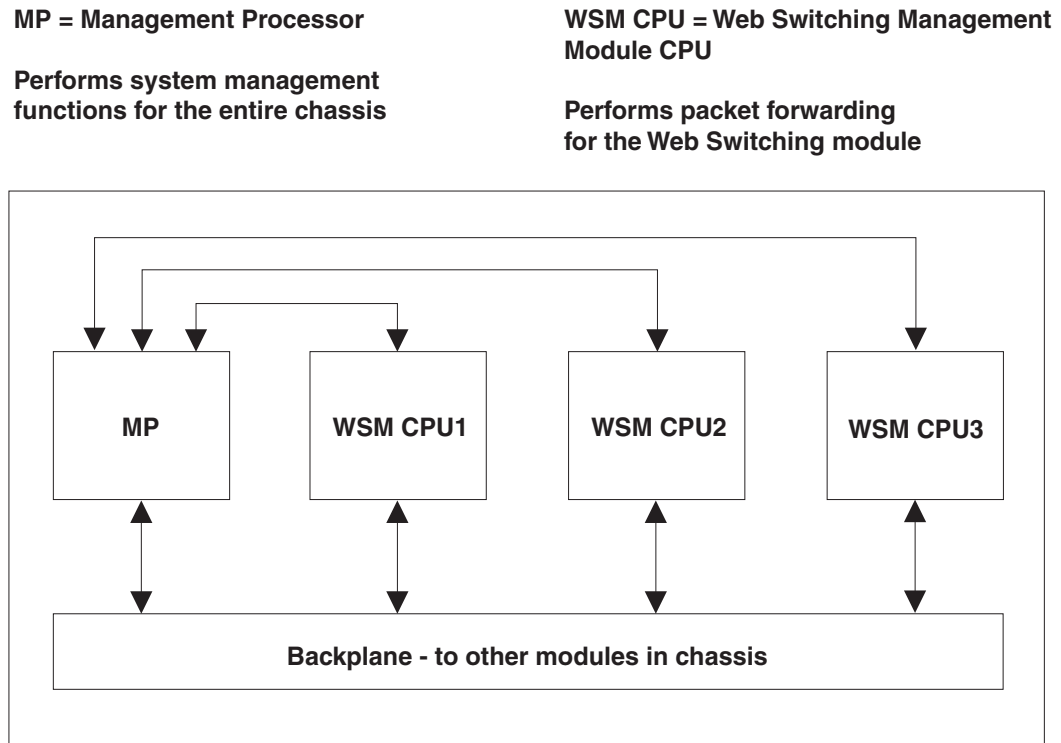
Figure 4.2 illustrates the architecture of the Web Switching Management Module.

---

**NOTE:** *By default, the WSM CPUs (the dedicated Layer 4 – 7 processors on the Web Switching Management Module) are disabled. To enable them, you must use the procedure in “Enabling the WSM CPUs” on page 4-5.*

---

**Figure 4.2 Architecture of Web Switching Management Module**



The Web Switching Management Module contains four processors. One of the processors is the Management Processor (MP) and contains the management software for the entire chassis. The other processors are web switching CPUs (WSM CPUs) and contain the Layer 2 – 7 packet processing code. The MP and each WSM CPU have their own flash memory with primary and secondary areas.

The Web Switching Management Module also contains a temperature sensor. The sensor generates a Syslog message and SNMP trap if the module's temperature exceeds a specified warning level or shutdown level. You can use the CLI or Web management interface to display the management module's temperature and to change the warning and shutdown temperature levels. See "Changing the Management Session from the MP to a WSM CPU" on page 4-9.

## WSM CPU Load Sharing

The Web Switching Management Module optimizes performance by distributing responsibility for the forwarding modules across the WSM CPUs, so that each WSM CPU has sole responsibility for a given forwarding module and the modules are as evenly distributed across the WSM CPUs in terms of bandwidth.

When you power on or reset the ServerIron 400 or ServerIron 800's Web Switching Management Module, the module assigns each of the forwarding modules to a WSM CPU according to each module's weight. A forwarding module's weight is a number that represents its total forwarding capacity. The weight is measured in units of 1 for each 100 Mbps. For example, Table 4.1 shows the weights for some common forwarding module types. Notice that the weight for 10/100 modules is based on the higher bandwidth (100 Mbps instead of 10 Mbps) for all ports.

**Table 4.1: Forwarding Module Weights**

Module type	Total Mbps capacity	Weight
24-port 10/100 Mbps	2400	24

**Table 4.1: Forwarding Module Weights (Continued)**

Module type	Total Mbps capacity	Weight
4-port 1000 Mbps	4000	40
8-port 1000 Mbps	8000	80

The device assigns the forwarding modules to WSM CPUs in numerical order (always starting with WSM CPU 1) and beginning with the module with the highest weight and working down to the module with the lowest weight.

The device assigns a forwarding module's ports to only one WSM CPU. A single module's ports are never distributed across multiple WSM CPUs.

The allocations determine the WSM CPU that will process traffic received on a forwarding module's ports. For example, if an 8-port Gigabit module in slot 3 is allocated to WSM CPU 1, then that CPU processes all the traffic received on the module's ports.

**NOTE:** If you hot-swap a module into or out of the chassis after the allocations have taken place at startup, the device does not re-allocate modules to even out the load sharing. Instead, the device allocates the module you insert to the WSM CPU that currently has the least weight allocated to it. If you remove a module, the device subtracts the module's weight from the WSM CPU to which the module was allocated.

Here are some examples of load sharing allocations for various configurations. Notice that for the ServerIron 400, each forwarding module is allocated to its own WSM CPU. The module's weights determine the WSM CPUs to which they are allocated. For the ServerIron 800, some WSM CPUs are allocated more than one module. Nonetheless, the allocations are based on the forwarding modules' weights and provide the most even distribution possible.

### Example Configuration 1

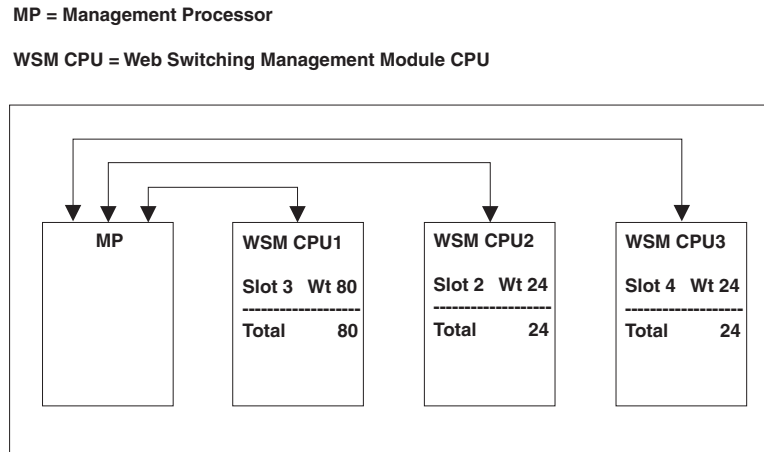
Table 4.2 shows a module configuration and the resulting WSM CPU allocations for a ServerIron 400. Notice that since the Web Switching Management Module does not have any forwarding ports, the module does not need to be allocated to a WSM CPU.

**Table 4.2: Example Configuration 1**

Slot	Module type	Weight	Order allocated	WSM CPU
1	Web Switching Management Module	n/a	n/a	n/a
2	24-port 10/100	24	2	WSM CPU 2
3	8-port Gigabit	80	1	WSM CPU 1
4	24-port 10/100	24	3	WSM CPU 3

Figure 4.3 shows the WSM CPU allocations for this configuration.

**Figure 4.3 WSM CPU allocations for example configuration 1**



The device begins with the highest-weight module, in this case the 8-port Gigabit module in slot 3, and allocates that module's ports to WSM CPU 1. The device then allocates the module with the second-highest weight, in this case the 24-port 10/100 module in slot 2, to the next WSM CPU with the lowest allocated weight, which is WSM CPU 2. Finally, the device allocates the last forwarding module, the 24-port 10/100 module in slot 4, to the next WSM CPU with the lowest allocated weight, WSM CPU 3.

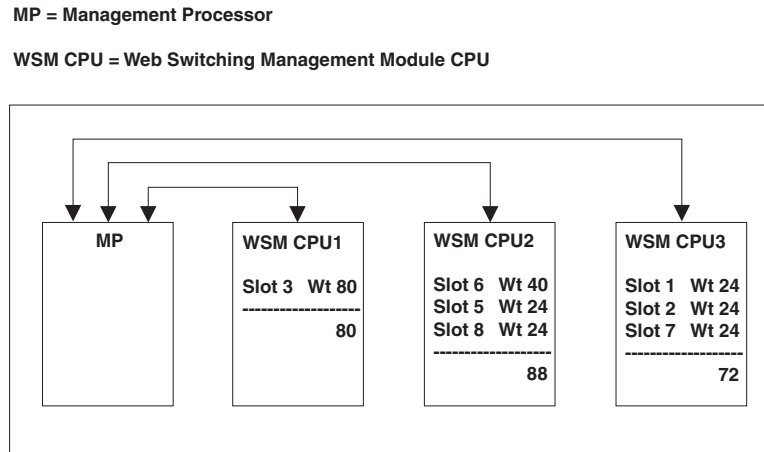
## Example Configuration 2

**Table 4.3: Example Configuration 2**

Slot	Module type	Weight	Order allocated	WSM CPU
1	24-port 10/100	24	3	WSM CPU 3
2	24-port 10/100	24	4	WSM CPU 3
3	8-port Gigabit	80	1	WSM CPU 1
4	Web Switching Management Module	n/a	n/a	n/a
5	24-port 10/100	24	5	WSM CPU 2
6	4-port Gigabit	40	2	WSM CPU 2
7	24-port 10/100	24	6	WSM CPU 3
8	24-port 10/100	24	7	WSM CPU 2

Figure 4.4 shows the WSM CPU allocations for this configuration.

**Figure 4.4 WSM CPU allocations for example configuration 2**



As in the previous example, the device starts with the highest-weight module, in this case the 8-port Gigabit module in slot 3, and allocates that module to WSM CPU 1. The device then allocates the second-highest weighted module to WSM CPU 2, and the third-highest weighted module to WSM CPU 3. For the next module, the device selects the WSM CPU with the lowest allocated weight; in this case, that is WSM CPU 3. And so on. As shown in this example, the resulting distribution is fairly even among the three CPUs.

## Displaying the Slot Allocations for the WSM CPUs

To display the allocations, enter the **show wsm-map** command. See “Determining the Slot Allocations for the WSM CPUs” on page 4-29.

## Changing Slot Allocations

The default allocations are applicable to almost all configurations. However, you can change the allocations if needed. See “Changing Slot Allocations for the WSM CPUs” on page 4-30.

## Enabling the WSM CPUs

Use the following procedure to enable the WSM CPUs.

---

**NOTE:** *By default, the WSM CPUs are disabled. To enable them, you must use the procedure in this section.*

---



---

**NOTE:** You do not need to use this procedure to enable the WSM CPUs if the configuration contains an FWLB policy for TCP traffic (for example, **ip policy 1 fw tcp 0 global**). The FWLB TCP policy also enables the WSM CPUs.

---

To enable the WSM CPUs, you must configure a policy by entering a command such as the following at the global CONFIG level of the CLI:

```
ServerIron(config)# ip policy 1 cache tcp 0 global
ServerIron(config)# write memory
```

**Syntax:** [no] ip policy <num> cache tcp 0 global

The <num> parameter specifies the policy number. If you have already configured a policy with the number 1, you can use another valid policy number. Otherwise, enter the command exactly as shown in this example. Enter the **write memory** command to save the configuration change to the startup-config file.

## Broadcasting Session Delete Messages to WSM CPUs

**NOTE:** The following section applies only to configurations where a client is connected to a router that is not the ServerIron's default gateway, and which is handled by a WSM CPU that does not also handle the ServerIron's default gateway.

When the ServerIron receives a DNS request from a client, the WSM CPU that handles the client creates a pair of session table entries. If the DNS server is connected to a module that is handled by a different WSM CPU, two pairs of session table entries are created: one pair by the client's WSM CPU, and one pair by the DNS server's WSM CPU. In this case, when the response is received from the DNS server, the server's WSM CPU deletes its session table entry pair and sends a session delete message to the WSM CPU that handles the ServerIron's default gateway. When the client is attached to the ServerIron's default gateway, this causes the client's session table entry pair to be deleted.

However, when the client is connected to a router that is not the ServerIron's default gateway, and which is handled by a WSM CPU that does not also handle the ServerIron's default gateway, the client's WSM CPU never receives the session delete message, and the client's session table entry pair is not deleted. In this instance, to delete the client's session table entry pair, you can configure the ServerIron to broadcast a session delete message to all of its WSM CPUs when it deletes the server's session table entry pair.

To do this, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# server udp-bc-client-session-del
```

**Syntax:** server udp-bc-client-session-del

## Upgrading the Software

If you need to upgrade the boot or flash code on the management processor (MP) or a WSM CPU, use the following procedures.

The MP and WSM CPUs run separate software. The MP runs chassis management software. The WSM CPUs run application-specific software, such as Layer 2 – 7 software. The procedures for upgrading MP and WSM CPUs are different.

---

**NOTE:** The MP and WSM CPU flash code must have the same version number. Otherwise, the WSM CPU functions are disabled. You can display the version numbers of the MP and WSM CPUs by entering the **show wsm-state** command. Also, if the version numbers are different, the command output displays a message.

---

---

**NOTE:** If you are upgrading from a TFTP server, make sure the chassis has network (IP) access to the server.

---

### Upgrading the MP Boot Code

To upgrade the MP boot code, use the same methods as for any other management module.

#### *USING THE CLI*

To upgrade MP boot code from a TFTP server, enter a command such as the following:

```
ServerIron# copy tftp flash 192.168.1.170 M2B07108.bin boot
```

**Syntax:** copy tftp flash <ip-addr> <image-file-name> boot

#### *USING THE WEB MANAGEMENT INTERFACE*

You cannot perform this procedure using the Web management interface.

### Upgrading the WSM CPU Boot Code

To upgrade the WSM CPU boot code, use the following CLI method.

### USING THE CLI

To upgrade WSM CPU boot code from a TFTP server, enter a command such as the following:

```
ServerIron# wsm copy tftp flash 192.168.1.170 W2B07100.bin boot
```

**Syntax:** wsm copy tftp flash <ip-addr> <image-file-name> boot

### USING THE WEB MANAGEMENT INTERFACE

You cannot perform this procedure using the Web management interface.

## Upgrading the MP Flash Code

To upgrade the MP flash code, use the same methods as for any other management module.

### USING THE CLI

To upgrade MP flash code (management software) from a TFTP server, enter a command such as the following:

```
ServerIron# copy tftp flash 192.168.1.170 wsm07200.bin primary
```

This command copies flash code from a TFTP server into the primary flash memory area for the MP. When you reload the software, the MP will boot the new code.

**Syntax:** copy tftp flash <ip-addr> <image-file-name> primary | secondary

To copy flash code from one flash memory area to the other, enter a command such as the following:

```
ServerIron# copy flash flash secondary
```

This command copies the flash code in the primary flash memory area to the secondary flash memory area for the MP.

**Syntax:** copy flash flash primary | secondary

The **primary** parameter copies the image in the secondary flash area to the primary flash area.

The **secondary** parameter copies the image in the primary flash area to the secondary flash area.

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Command in the tree view to expand the list of command options.
3. Click on the plus sign next to TFTP under Command in the tree view to expand the list of TFTP options.
4. Select the Image link to display the TFTP Image panel.
5. Enter the address of the TFTP server in the TFTP Server IP field.
6. Enter the image file name in the Image File Name field.

---

**NOTE:** The TFTP client on the Foundry device supports only 8.3 format file names (up to eight characters in the name plus up to three characters in the extension). Make sure that if you rename the file on your TFTP server, you give the file a name that conforms to these rules.

---

7. Specify the destination of the image file you are transferring by selecting Primary or Secondary next to Flash.
8. Click on the Copy from Server button to start the file transfer.

## Upgrading the WSM CPU Flash Code

To upgrade the WSM CPUs, use the following CLI method.

### USING THE CLI

To upgrade the WSM CPUs, enter a command such as the following at the Privileged EXEC level of the CLI:

```
ServerIron# wsm copy tftp flash 109.157.22.26 wsp07200.bin primary
```

This command upgrades the WSM CPUs by copying a flash code image from a TFTP server to the primary flash for each of the WSM CPUs on the module.

To copy the flash code from the primary flash to the secondary flash for each of the WSM CPUs on the module, enter a command such as the following:

```
ServerIron# wsm copy flash flash secondary
```

**Syntax:** wsm copy tftp flash <tftp-server-ip-addr> <image-file-name> primary | secondary

**Syntax:** wsm copy flash flash primary | secondary

The **primary** and **secondary** parameters identify either the primary or secondary flash on the WSM CPUs. For each command, the parameter specifies the destination of the copy operation.

---

**NOTE:** The **slot1** | **slot2** parameter is not supported in this release.

---

#### *USING THE WEB MANAGEMENT INTERFACE*

This procedure is not supported in the Web management interface.

## Changing the Default Boot Source

By default, the Web Switching Management Module's processors boot from the primary flash areas on the module. Each processor boots from its own primary flash. The MP boots first, then the WSM CPUs boot.

You can change the default boot source to one of the following:

- Primary flash (the default)
- Secondary flash
- Interactive

The interactive option pauses during bootup of the WSM CPUs to allow you to select the boot source for the WSM CPUs. You must use this method if you want to boot the WSM CPUs from a TFTP server. Otherwise, this method is used for troubleshooting, such as when the previous TFTP transfer attempt aborts prematurely.

To change the default boot source, use one of the following methods.

#### *USING THE CLI*

To change the default boot source, enter commands such as the following at the global CONFIG level of the CLI:

```
ServerIron(config)# wsm boot secondary
ServerIron(config)# write memory
```

This command configures the module to boot from the secondary flash by default.

---

**NOTE:** The **write memory** command saves the change to the startup-config file. You must save the configuration change for the change to remain in effect after you reboot.

---

**Syntax:** wsm boot primary | secondary | interactive

The **primary** and **secondary** parameters specify a flash memory location. The **interactive** parameter causes the device to pause during bootup to allow you to specify the boot source for the WSM CPUs. You must use this method if you want to boot the WSM CPUs from a TFTP server. Otherwise, the **interactive** parameter is used for troubleshooting.

To configure the module to pause during booting to allow you to specify the boot source, enter the following command:

```
ServerIron(config)# wsm boot interactive
```



After you set the boot source to interactive and reboot, enter a command such as the following at the Privileged EXEC level of the CLI to boot the WSM CPUs:

```
ServerIron# wsm boot tftp 192.168.1.170 wsp07200.bin
```

This command copies the WSM CPU flash code image from the specified TFTP server to a WSM CPU address space from which the WSM CPU can boot.

**Syntax:** wsm boot primary | secondary | tftp <ip-addr> <image-file-name>

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to System.
4. Select the [Boot Sequence](#) link to display the Boot Sequence List panel.
5. Select the primary boot source by clicking on the radio button next to the name.

---

**NOTE:** You cannot select the interactive option using the Web management interface. To select this option, use the CLI.

---

6. To specify a secondary boot source, go to step 5. The device tries the boot sources in the order you specify them.
7. Select Add to add the change to the device's running-config.
8. If you want the change to remain in effect following the next system reload, select the [Save](#) link to save the configuration change to the startup-config file.

## Changing the Management Session from the MP to a WSM CPU

By default, management sessions you open with the Web Switching Management Module are established with the MP. However, you can establish a session directly with a WSM CPU. Each WSM CPU supports some commands at the Privileged EXEC level.

---

**NOTE:** You can enter configuration commands only to the MP, not directly to a WSM CPU.

---

The CLI provides a remote login facility for changing the management session to a WSM CPU. When you log in to a WSM CPU, the CLI management session changes from the MP to the WSM CPU. At this point, commands apply only to the WSM CPU. To enter commands to the MP, you must log out of the WSM CPU. The CLI prompt changes to indicate the chassis slot number and WSM CPU you are logged on to.

### Logging In to a WSM CPU

To log in to a WSM CPU, enter a command such as the following at the Privileged EXEC level of the CLI:

```
ServerIron# rconsole 2 1
ServerIron2/1 #
```

This command changes the management session from the MP to WSM CPU 1 on the Web Switching Management Module in slot 2. Notice that the end of the command prompt changes to indicate the slot number and WSM CPU number.

**Syntax:** rconsole <slotnum> <cpunum>

The <slotnum> parameter specifies the chassis slot that contains the module.

- Slots on a four-slot chassis are numbered 1 – 4, from top to bottom.
- Slots on an eight-slot chassis are numbered 1 – 8, from left to right.

The <cpunum> parameter specifies the WSM CPU. The WSM CPUs are numbered from 1 – 3.

## Entering Commands to the WSM CPU

You can enter a command at the WSM CPU's CLI prompt. Here is an example:

```
ServerIron2/1 # show server real
ServerIron2/1 #Real Servers Info

Name : r33                               Mac-addr: 0001.0246.5f0c
IP:207.95.6.33      Range:1      State:Active      Wt:0      Max-conn:1000000

Port      State      Ms  CurConn  TotConn  Rx-pkts  Tx-pkts  Rx-octet  Tx-octet  Reas
----      -
http      active      0  0        0        0        0        0        0        0
tftp      active      0  0        0        0        0        0        0        0
telnet    active      0  0        0        0        0        0        0        0
default   unbnd       0  0        0        0        0        0        0        0

Server    Total      0        0        0        0        0        0        0
```

This example shows output from the **show server real** command. The command shows the real server information contained on this particular WSM CPU.

See “WSM CPU Commands” on page 4-10 for a list of the commands you can enter on a WSM CPU.

## Logging Out from the WSM CPU

To log out from a management session with a WSM CPU, enter the following command at the Privileged EXEC level of the CLI:

```
ServerIron2/1 # rconsole-exit
ServerIron#
```

**Syntax:** rconsole-exit

## WSM CPU Commands

The following commands are supported on each WSM CPU:

- **clear server traffic** – Clears traffic statistics for servers
- **exit** – Exits the Privileged EXEC mode
- **wsm** – Displays all the **wsm** commands
- **quit** – Exits to the User EXEC level of the CLI
- **rel-msg** – Enable display of messages exchanged by the MP and this WSM CPU. This is for diagnostic use only.
- **show** – Displays system information

---

**NOTE:** Only some **show** commands are supported on individual WSM CPUs. See the list below.

---

- **write** – Saves the running-config to the startup-config file

For information about any of these commands except **wsm** and **rel-msg**, see the *Foundry ServerIron Command Line Interface Reference*. The **wsm** commands are described in other sections of this chapter (“Using the Web Switching Management Module”). The **rel-msg** command is used by Foundry Networks for troubleshooting and is not described in detail in the user documentation.

The following **show** commands are supported on the WSM CPUs:

- **arp** – Displays the ARP table
- **cache-group** – Displays Transparent Cache Switching (TCS) cache group information
- **configuration** – Displays the startup-config file
- **fw-group** – Displays firewall group information for Firewall Load Balancing (FWLB)
- **ip** – Displays the IP information for the device, including the management address and the default gateway
- **mac-address** – Displays the MAC address table
- **rel-msg** – Displays messages exchanged by the MP and this WSM CPU
- **running-config** – Displays the running-config
- **server** – Displays Layer 4 – 7 server information.
- **version** – Displays software version information as well as some basic status information
- **vlan** – Displays VLAN information

For information about any of the **show** commands except **rel-msg**, see the *Foundry ServerIron Command Line Interface Reference*. The **rel-msg** command is used by Foundry Networks for troubleshooting and is not described in detail in the user documentation.

## Temperature Sensor

The Web Switching Management Module contains a temperature sensor. Depending on the temperature reported by the sensor, the software can send a warning if the temperature exceeds the normal threshold and can even shut the module down if the temperature exceeds the safe threshold. The software reads the temperature sensor according to the chassis poll time, which is 60 seconds by default.

When the software reads the temperature sensor, if the temperature equals or exceeds the warning or shutdown temperature, the software does the following:

- **Warning message** – If the temperature of the module reaches the warning value, the software sends a Syslog message to the Syslog buffer and also to the SyslogD server, if configured. In addition, the software sends an SNMP trap to the SNMP trap receiver, if you have configured the device to use one.
- **Shutdown** – If the temperature matches or exceeds the shutdown temperature, the software sends a Syslog message to the Syslog buffer and also to the SyslogD server if configured. The software also sends an SNMP trap to the SNMP trap receiver, if you have configured the device to use one.

If the temperature equals or exceeds the shutdown temperature for five consecutive polls of the temperature by the software, the software shuts down the module to prevent damage.

You can display the temperature of the module. You also can change the warning and shutdown temperatures and the chassis poll time.

## Displaying the Temperature

By default, the software polls the temperature sensor on the module every 60 seconds to get the current temperature. This poll rate is controlled by the chassis poll time, which also controls how often the software polls other system components. You can display the temperature of the module using either of the following methods.

### *USING THE CLI*

To display the temperature of a Web Switching Management Module, enter the following command at any level of the CLI:

```
ServerIron> show chassis
power supply 1 not present
power supply 2 not present
power supply 3 ok
power supply 4 not present
power supply 1 to 4 from bottom to top
fan 1 ok
fan 2 bad
fan 3 ok
fan 4 ok
Current temperature : 34.5 C degrees
Warning level : 45 C degrees, shutdown level : 55 C degrees
```

**Syntax:** show chassis

### *USING THE WEB MANAGEMENT INTERFACE*

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to display the monitoring options.
3. Select the Device link to display the Device Information panel. The temperature is listed in the Temperature field. The temperature information is color coded to indicate the state.
  - Green indicates the temperature is within the normal operating range.
  - Orange indicates the temperature has reached the warning level.
  - Red indicates the temperature has reached the shutdown level.

---

**NOTE:** You also can display the Device Information panel by clicking on the graphic of the chassis panel, in the upper right frame. The graphic is shown only if the Web management interface frames are enabled.

---

## **Displaying Temperature Messages**

The software sends a Syslog message and an SNMP trap if the temperature crosses the warning or shutdown thresholds. The following methods describe how to view the system log on the device. If you have configured the device to use a SyslogD server or SNMP trap receiver, see the documentation for the server or receiver.

### USING THE CLI

To display the system log, enter the following command at any CLI level:

```
ServerIron# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
Buffer logging: level ACDMEINW, 8 messages logged
level code: A=alert C=critical D=debugging M=emergency E=error
I=informational N=notification W=warning

Static Log Buffer:

Dynamic Log Buffer (50 entries):

at 0 days 0 hours 2 minutes 0 seconds, level alert
Temperature 48.0 C degrees, warning level 45.0 C degrees, shutdown level 55.0 C
degrees

at 0 days 0 hours 1 minutes 0 seconds, level alert
Temperature 50.0 C degrees, warning level 45.0 C degrees, shutdown level 55.0 C
degrees
```

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to display the Monitor options.
3. Select the System Log link to display the system log.

## Changing Temperature Warning and Shutdown Levels

The default warning temperature is 45.0 C degrees. The default shutdown temperature is 55.0 C degrees. You can change the warning and shutdown temperatures using the following commands. The valid range for each value is 0 – 125 C degrees.

---

**NOTE:** You cannot set the warning temperature to a value higher than the shutdown temperature.

---

### USING THE CLI

To change the temperature at which the module sends a warning, enter a command such as the following at the Privileged EXEC level of the CLI:

```
ServerIron# temperature warning 47
```

**Syntax:** temperature warning <value>

The <value> can be 0 – 125.

To change the shutdown temperature, enter a command such as the following at Privileged EXEC level of the CLI:

```
ServerIron# temperature shutdown 57
```

**Syntax:** temperature shutdown <value>

The <value> can be 0 – 125.

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.

2. Select the [Advance](#) link to display the following panel.

**System**

Tag Type:	8100
Broadcast Limit:	0
Switch Age Time:	300
Default VLAN ID:	1
Chassis Poll Interval (sec):	60
Temperature Warning Threshold(C):	45
Temperature Shutdown Threshold(C):	55
Gig Port Default:	Neg-Full-Auto
Mirror Slot:	None
Port:	None

[\[Home\]](#)
[\[Site Map\]](#)
[\[Logout\]](#)
[\[Save\]](#)
[\[Frame Enable\]](#)
[\[Disable\]](#)
[\[TELNET\]](#)

3. Edit the value in the Temperature Warning Threshold field to change the warning temperature.
4. Edit the value in the Temperature Shutdown Threshold field to change the shutdown temperature.
5. Click the Apply button to send the configuration change to the active module's running-config file.
6. If you want the change to remain in effect following the next system reload, select the [Save](#) link to save the configuration change to the startup-config file.

## Changing the Chassis Polling Interval

The software reads the temperature sensor and polls other hardware sensors according to the value set for the chassis poll time, which is 60 seconds by default. You can change chassis poll time using the CLI.

### USING THE CLI

To change the chassis poll time, enter a command such as the following at the global CONFIG level of the CLI:

```
ServerIron(config)# chassis poll-time 200
```

**Syntax:** chassis poll-time <value>

The <value> can be 0 – 65535.

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.

2. Select the [Advance](#) link to display the following panel

**System**

Tag Type:	<input type="text" value="8100"/>
Broadcast Limit:	<input type="text" value="0"/>
Switch Age Time:	<input type="text" value="300"/>
Default VLAN ID:	<input type="text" value="1"/>
Chassis Poll Interval (sec):	<input type="text" value="60"/>
Temperature Warning Threshold(C):	<input type="text" value="45"/>
Temperature Shutdown Threshold(C):	<input type="text" value="55"/>
Gig Port Default:	<input type="text" value="Neg-Full-Auto"/>
Mirror Slot:	<input type="text" value="None"/> <input type="text" value="Port: None"/>

[\[Home\]](#)
[\[Site Map\]](#)
[\[Logout\]](#)
[\[Save\]](#)
[\[Frame Enable\]](#)
[\[Disable\]](#)
[\[TELNET\]](#)

3. Edit the value in the Chassis Poll Interval field to change polling interval. You can enter a value from 0 – 65535. The default is 60 seconds.
4. Click the Apply button to send the configuration change to the active module's running-config file.
5. If you want the change to remain in effect following the next system reload, select the [Save](#) link to save the configuration change to the startup-config file.

## Configuring Redundancy

The ServerIron 400 and ServerIron 800 support management module redundancy. You can install two management modules in the chassis and use them as an active-standby pair. If the active management module goes down, the standby module takes over, continuing the operation of the device.

### Switchover

When you power on or reload a Chassis device that contains two redundant management modules, the active redundant management module is selected based on the chassis slot previously specified by you or according to the lower slot number.

After the active module is selected, the active module loads its boot and flash code (boot and system software) and its system-config file and manages the system. The standby module also boots, using its own boot code but using the active module's flash code and system-config file. The standby module monitors the heartbeat of the active module. If the active module becomes unavailable, the standby module notices the absence of the heartbeat and assumes management control of the system.

**NOTE:** By default, the system does not use the boot code on the active module to boot the standby module. If you upgrade the boot code on the active module and the code contains a problem, you can still use the system by running the older boot code that is on the standby module. If desired, you can configure the standby to synchronize with the active module's boot code. See "File Synchronization Between the Active and Standby Redundant Management Modules" on page 4-19.

The standby module's system-config file is updated whenever the system-config file on the active module is updated. In addition, the running-config file on the standby module is updated at regular intervals to match the active module's running-config data. Thus, when a switchover occurs, the standby module also can reinstate the configuration data in the active module's running-config.

Following this switchover to the standby module, the standby module becomes the active module and continues to manage the system. When the other redundant management module (the one that used to be the active module) becomes available again or is replaced, that module becomes the standby module.

The active module also monitors the standby module. If the standby module becomes unavailable, the active module tries to reboot the standby module. You can display the status of each module using the CLI, as described in “Determining Redundant Management Module Status” on page 4-18.

## Management Sessions

You can establish management sessions only with the active redundant management module, not with the standby redundant management module. During switchover, all the management sessions open on the system are closed. To manage the system following a switchover, you must open a new management session. Although the system's MAC addresses change following switchover, the IP addresses do not. You can open new management sessions on the same IP addresses you were using before the switchover if desired.

To establish a serial connection to the CLI, you must move the serial cable to the serial port on the active redundant management module.

## Syslog and SNMP Traps

When a switchover occurs, the software sends a Syslog message to the local Syslog buffer and also to the SyslogD server, if you have configured the device to use one. In addition, if you have configured an SNMP trap receiver, the software sends an SNMP trap to the receiver.

When the system is powered on or otherwise reset normally, the software sends a cold start message and trap. However, if the system is reset as the result of switchover to the standby redundant management module, the software instead sends a switchover message and trap.

## MAC Address Changes

The MAC addresses in the system are based on the MAC address of the active management module. During switchover, the system's MAC addresses change and the system sends out gratuitous ARP requests to flush the old MAC addresses from the ARP caches on attached IP devices, and update the caches with the Foundry device's new MAC addresses.

## Configuring the Redundant Management Parameters

You can configure the following redundant management module parameters:

- Installation parameters:
  - Slot configuration. As with other module types, you must configure a chassis slot for the type of module you are installing in the slot.
  - Active redundant management module slot. By default, the redundant management module with the lower slot number is the active module.
- Operational parameters:
  - Boot code synchronization. By default, the standby redundant management module does not automatically synchronize to the boot code version installed on the active module. The standby module does automatically synchronize to the flash code (system software) on the active module.
  - Synchronization interval for running-config file
  - Warning and shutdown temperatures

## Installing Redundant Management Modules

To install a redundant management module, perform the following tasks:

- Configure the chassis slot to receive the module.
- Insert the module.



- Specify the default active module (if you do not want to use the system default, which is the redundant management module with the lower slot number).

In addition, if you use a TFTP or BootP server to boot the active module, you need to copy the flash code (system software) into the primary or secondary flash on the active redundant management module, then direct the active redundant management module to use the code to boot the standby module.

A standby redundant management module does not boot from a TFTP or BootP server.

### Configuring the Chassis to Receive the Module

When you plan to insert a module into a chassis slot, you first must configure the slot to receive the module unless the slot already contains the same type of module.

**Syntax:** module <slot-num> <module-type>

The <slot-num> parameter specifies the chassis slot to contain the module:

- Slots in a 4-slot chassis are numbered 1 – 4, from top to bottom.
- Slots in an 8-slot chassis are numbered 1 – 8, from left to right.

The <module-type> parameter specifies the platform and port configuration of the redundant management module.

### Specifying the Default Active Module

By default, the redundant management module in the lower slot number becomes the active redundant management module when you start the system. For example, if you install redundant management modules in slots 1 and 8 in an 8-slot chassis, the default active module is the module in slot 1.

---

#### NOTE:

- Slots on a four-slot chassis are numbered 1 – 4, from top to bottom.
  - Slots on an eight-slot chassis are numbered 1 – 8, from left to right.
- 

You can override the default and specify the active module.

---

**NOTE:** The change does not take effect until you reload the system. If you save the change to the active module's system-config file before reloading, the change persists across system reloads. Otherwise, the change affects only the next system reload.

---

To override the default and specify the active redundant management module, enter the following commands:

```
BigIron(config)# redundancy
BigIron(config-redundancy)# active-management 5
```

**Syntax:** active-management <slot-num>

The <slot-num> parameter specifies the chassis slot:

- Slots in a 4-slot chassis are numbered 1 – 4, from top to bottom.
- Slots in an 8-slot chassis are numbered 1 – 8, from left to right.

This command overrides the default and makes the redundant management module in slot 5 the active module following the next reload. The change affects only the next reload and does not remain in effect for future reloads.

To make the change permanent across future reloads, enter the **write memory** command to save the change to the startup-config file, as shown in the following example:

```
ServerIron(config)# redundancy
ServerIron(config-redundancy)# active-management 5
ServerIron(config-redundancy)# write memory
```

---

**NOTE:** If you do not save the change to the startup-config file, the change affects only the next reload.

---

## Inserting the Module

You can remove and insert modules when the system is powered on. Make sure you adhere to the cautions noted in the "Installation Precautions" section of the "Installing a Foundry Layer 2 Switch or Layer 3 Switch" chapter in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

1. Put on an ESD wrist strap and attach the clip end to a metal surface (such as an equipment rack) to act as ground.
2. Remove the module or faceplate from the slot:
3. If you are replacing another module, loosen the two screws on the module you are removing.
  - Pull the module ejectors towards you, away from the module front panel. The module will unseat from the backplane.
  - Pull the module out of the chassis and place in an anti-static bag for storage.
4. If you are installing a redundant management module in an unoccupied module slot, remove the blank faceplate from the slot in which the module is to be installed. Place the blank faceplate in a safe place for future use.
5. Remove the redundant management module from its packaging.
6. Insert the module into the chassis slot and glide the module along the module guide until the module ejectors on the front of the module touch the chassis.
  - Modules for 4-slot chassis slide in horizontally with the module label on the left.
  - Modules for 8-slot chassis slide in vertically with the module label at the top.
7. Push the ejectors toward the center of the module until they are flush with the front panel of the module. The module will be fully seated in the backplane.
8. Tighten the two screws at either end of the module.
9. If you do not use one or more of the slots, make sure that a slot faceplate is still attached over each unused slot for safe operation and proper system cooling.

## Determining Redundant Management Module Status

You can determine the status of a redundant management module in the following ways:

- Status LED – The redundant management module has two green LEDs on the right side of the CLI serial port. The lower LED shows the management status.
- Module information in software – The module information displayed by the software indicates whether the module is the active module, the standby module, or has another status.

### Status LED

If you are located near the device, you can determine which redundant management module is currently the active module and which one is the standby by observing the upper green LED to the right of the serial management port. If the upper green LED is lit, the module is currently the active redundant management module. If the LED is dark, the module is the standby. The lower green LED indicates the power status. If the lower LED is dark, the module is not receiving power. (A module without power will not function as the active or standby module.)

### Software

You can display status information for the modules using either of the following methods.

---

#### NOTE:

- Slots on a four-slot chassis are numbered 1 – 4, from top to bottom.
  - Slots on an eight-slot chassis are numbered 1 – 8, from left to right.
-

To display the status of a redundant management module using the CLI, enter the following command at any CLI level:

**Syntax:** show module

---

**NOTE:** The module descriptions do not distinguish between SX and LX ports.

---

The Status column shows the module status. The redundant management modules can have one of the following statuses:

- **ACTIVE** – The module is currently the active management module.
- **STANDBY** – The module is the standby management module.
- **COMING UP** – The module is coming up as the standby module. This status can be observed during switchover.

The statuses above apply only to management modules. The following statuses apply only to host modules:

- **FAILED** – This status applies only to host modules, not to management modules. This status indicates that the host module failed to come up.
- **OK** – This status applies only to host modules, not to management modules. This status indicates that the module came up and is operating normally.

## Displaying Switchover Messages

You can determine whether a switchover has occurred by viewing the system log or the traps logged on an SNMP trap receiver.

To view the system log, enter the **show logging** command. The following message indicates a failover has occurred.

Message Level	Message	Explanation
Alert	Management module at slot <slot-num> state changed from <module-state> to <module-state>.	<p>Indicates a state change in a management module.</p> <p>The &lt;slot-num&gt; indicates the chassis slot containing the module.</p> <p>The &lt;module-state&gt; can be one of the following:</p> <ul style="list-style-type: none"> <li>• active</li> <li>• standby</li> <li>• crashed</li> <li>• coming-up</li> <li>• unknown</li> </ul>

## File Synchronization Between the Active and Standby Redundant Management Modules

Each redundant management module contains four files that can be synchronized between the two modules:

- **Boot code** – The code the module runs when it first starts up. By default, the boot code is not synchronized between redundant management modules. This ensures that the system can still operate if a new version of

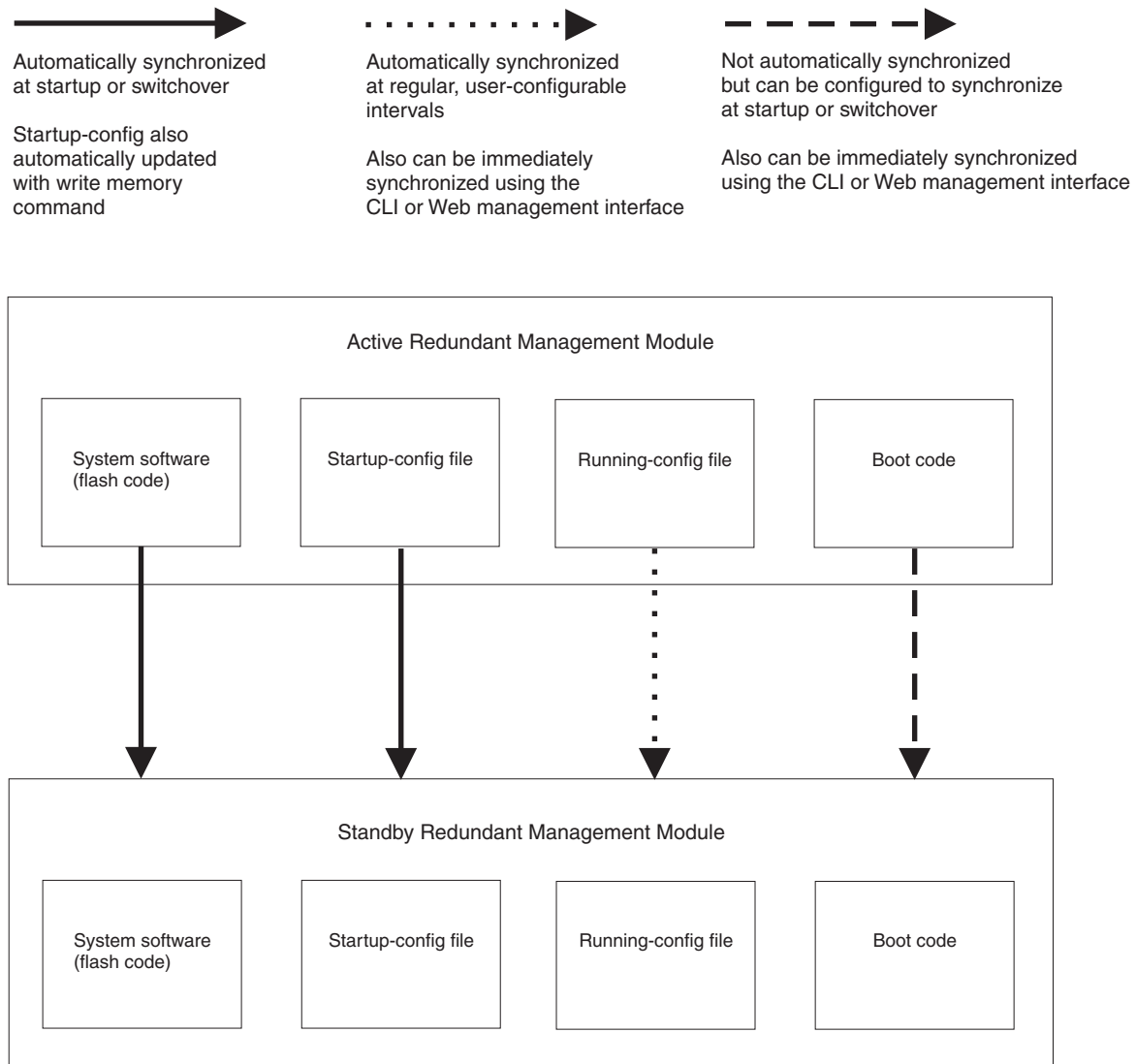
boot code contains a bug that prohibits normal operation. If the new code on the active module does not work properly, the system can still run using the older version of boot code on the standby module.

You can configure the standby redundant management module to synchronize with the active redundant management module's boot code whenever the boot code on the active module is updated or the system starts up.

- **Flash code (system software)** – The flash code is automatically synchronized between the redundant management modules. When the system starts up, the active redundant management module sends its flash code to the standby redundant management module to boot the module.
- **System-config file** – The system-config file is automatically copied from the active redundant management module to the standby redundant management module when the system starts up. The file is also copied to the standby module whenever you save changes to the file. If switchover occurs, the standby redundant management module loads system parameters from the running-config data that was last received from the active redundant management module. If the standby module did not receive running-config data from the active module, the standby module uses configuration information in the system-config file copied from the active module.
- **Running-config** – The running-config is automatically copied from the active redundant management module to the standby redundant management module at regular intervals. The default interval is 10 seconds. You can change the interval to 4 – 20 seconds. If you set the interval to 0, the configuration data is not copied to the standby redundant management module. As described above, if switchover occurs, the standby redundant management module loads system parameters from the running-config that was last received from the active redundant management module.

Figure 4.5 shows how the files are synchronized between the active redundant management module and the standby redundant management module.

**Figure 4.5 Redundant management module file synchronization**



### Displaying the Synchronization Settings

You can independently synchronize the following types of software between the active and standby modules:

- boot code
- flash code (system software)
- startup-config file
- running-config

When you synchronize software between the modules, the active module copies its software to the standby module.

To display the current file synchronization settings, enter the following command:

```
ServerIron# sync-standby
```

```

Sync code image: TRUE
Sync config data: TRUE
Sync boot image: FALSE
Running-config sync interval is 10 seconds

```

**NOTE:** The values shown in this example are the default values.

**Syntax:** sync-standby

**NOTE:** The **sync-standby** command has optional parameters. If you enter one of the parameters, the CLI synchronizes software between the modules. To display the synchronization settings instead of synchronizing software, enter the command without parameters.

This display shows the following information.

**Table 4.4: CLI Display of Synchronization Settings**

This Field...	Displays...
Sync code image	<p>Indicates whether the active module is configured to automatically synchronize its flash code with the standby module. The value can be one of the following:</p> <ul style="list-style-type: none"> <li>FALSE – The code is not automatically synchronized.</li> <li>TRUE – The code is automatically synchronized.</li> </ul>
Sync config data	<p>Indicates whether the active module is configured to automatically synchronize its startup-config file with the standby module. The value can be one of the following:</p> <ul style="list-style-type: none"> <li>FALSE – The startup-config file is not automatically synchronized.</li> <li>TRUE – The startup-config file is automatically synchronized.</li> </ul>
Sync boot image	<p>Indicates whether the active module is configured to automatically synchronize its boot code with the standby module. The value can be one of the following:</p> <ul style="list-style-type: none"> <li>FALSE – The boot code is not automatically synchronized.</li> <li>TRUE – The boot code is automatically synchronized.</li> </ul>
Running-config sync interval	<p>Indicates whether the active module is configured to automatically synchronize its running-config with the standby module. The value can be one of the following:</p> <ul style="list-style-type: none"> <li>FALSE – The running-config is not automatically synchronized.</li> <li>TRUE – The running-config is automatically synchronized.</li> </ul>

### Immediately Synchronizing Software

You can immediately synchronize software between the active and standby management modules. When you synchronize software, the active module copies the software you specify to the standby module, replacing the software on the standby module.

To immediately synchronize the boot code on the standby module with the boot code on the active module, enter the following command at the Privileged EXEC level of the CLI:

```
ServerIron# sync-standby boot
```

**Syntax:** sync-standby boot

To immediately synchronize the flash code (system software) on the standby module with the boot code on the active module, enter the following command at the Privileged EXEC level of the CLI:

```
ServerIron# sync-standby code
```

**Syntax:** sync-standby code

To immediately synchronize the running-config on the standby module with the running-config on the active module, enter the following command at the Privileged EXEC level of the CLI:

```
ServerIron# sync-standby running-config
```

**Syntax:** sync-standby running-config

To immediately synchronize the startup-config file on the standby module with the startup-config file on the active module, enter the following command at the Privileged EXEC level of the CLI:

```
ServerIron# sync-standby startup-config
```

**Syntax:** sync-standby startup-config

### Automating Synchronization of Software

Automatic synchronization of the flash code, running-config, and system-config file is enabled by default. Automatic synchronization of the boot code is disabled by default.

The CLI commands for automating synchronization of software between the active and standby modules is the same as the syntax for immediately synchronizing the software. The only difference is the CLI level where you enter the commands.

- To immediately synchronize software, enter the command at the Privileged EXEC level.
- To automate synchronization starting with the next software reload or system reset and each reload or reset after that, enter the command at the Redundancy CONFIG level.

Automatic synchronization of the flash code, running-config, and system-config file is enabled by default. Automatic synchronization of the boot code is disabled by default. To change the automatic synchronization setting, use one of the following commands:

**Syntax:** [no] sync-standby boot

**Syntax:** [no] sync-standby code

**Syntax:** [no] sync-standby startup-config

**Syntax:** [no] sync-standby running-config [<num>]

To disable automatic synchronization of the boot code, flash code, or startup-config file, enter “no” in front of the command.

The <num> parameter with the **sync-standby running-config** command specifies the synchronization interval. You can specify from 4 – 20 seconds. The default is 10 seconds.

To disable automatic synchronization of the running-config, set the synchronization interval (the <num> parameter) to 0.

### Switching Over to the Standby Redundant Management Module

If you reload the software using the **reload** command, the behavior of the management modules is the same as when you power the system on. The system selects the active module based on the slot you specified or based on the lower slot number if you did not specify a slot. Then both redundant management modules load their own boot code and load the active redundant management module's flash code (system software) and system-config file.

If you do not want to reload the system but you instead want to force the system to switch over to the standby module (and thus make it the active redundant management module), use one of the following methods.

To switch over to the other redundant management module, enter a command such as the following:

```
ServerIron# reset 2
```

**Syntax:** reset <slot-num>

Specify the slot number containing the currently active management module. Do not specify the slot number containing the standby module to which you want to switch over.

The <slot-num> parameter specifies the chassis slot:

- Slots in a 4-slot chassis are numbered 1 – 4, from top to bottom.
- Slots in an 8-slot chassis are numbered 1 – 8, from left to right.

## Displaying Web Switching Management Module Information

You can display the following Web Switching Management Module information:

- Software versions – see “Displaying the Software Version Running on the Module” on page 4-24
- General module information – “Displaying General Module Information” on page 4-25
- Module status – see “Determining Module Status” on page 4-26
- Slot allocations for the WSM CPUs – see “Determining the Slot Allocations for the WSM CPUs” on page 4-29

### Displaying the Software Version Running on the Module

To display the software version running on the Web Switching Management Module, use either of the following methods.



## USING THE CLI

To display the software version running on the module, enter the following command at any CLI level:

```

ServerIron(config)# show version
  SW: Version 07.2.00T52 Copyright (c) 1996-1999 Foundry Networks, Inc.
      Compiled on Sep 25 2000 at 21:20:39 labeled as mp
  HW: Chassis 8000 Router, SYSIF version 21
=====
SL 3: B24E Copper Switch Module
  2048 KB BRAM, SMC version 2, ICBM version 21
  256 KB PRAM(256K+0K) and 2048*8 CAM entries for DMA 8, version 0808
  256 KB PRAM(256K+0K) and shared CAM entries for DMA 9, version 0808
  256 KB PRAM(256K+0K) and shared CAM entries for DMA 10, version 0808
=====
SL 4: B24E Copper Switch Module
  2048 KB BRAM, SMC version 2, ICBM version 21
  256 KB PRAM(256K+0K) and 2048*8 CAM entries for DMA 12, version 0808
  256 KB PRAM(256K+0K) and shared CAM entries for DMA 13, version 0808
  256 KB PRAM(256K+0K) and shared CAM entries for DMA 14, version 0808
=====
SL 6: B0GMR WSM Management Module, WSM, ACTIVE
  0 MB SHM, 3 Application Processors
8192 KB BRAM, SMC version 1, ICBM version 21
SW: (1)07.2.00T71 (2)07.2.00T71 (3)07.2.00T71
=====
Active management module:
  466 MHz Power PC processor 750 (version 8/8300) 62 MHz bus
  512 KB boot flash memory
  8192 KB code flash memory
  256 KB SRAM
  256 MB DRAM
The system uptime is 1 hours 54 minutes 2 seconds

```

**Syntax:** show version

The command shows all the software versions running on the device. The Web Switching module information is shown in this example in bold text.

## USING THE WEB MANAGEMENT INTERFACE

You cannot display the module software versions using the Web management interface.

## Displaying General Module Information

To display general module information, use the following method.

### USING THE CLI

To display general information for a Web Switching Management Module, enter the following command at any CLI level:

```
ServerIron(config)# show wsm-state
=====
WSM MODULE (6) App CPU      0 MB SHM, 3 Application Processors
      CPU 0 in state of WSM_STATE_RUNNING
      CPU 1 in state of WSM_STATE_RUNNING
      CPU 2 in state of WSM_STATE_RUNNING
-----
Module 6 App CPU 1, SW: Version 07.2.00T71
Compiled on Sep 25 2000 at 21:33:50 labeled as wsm-cpu3b
DRAM 268M, BRAM 262K, FPGA Version 0050
Code Flash 4M: Primary (880346 bytes, 07.2.00T71),
                Secondary (871842 bytes, 07.0.00T71)
Boot Flash 131K, Boot Version 06.00.00
The system uptime is 0 day 1 hour 54 minute 17 second
General Status: 0 ipc msg rec, 2 ipc msg sent
-----
Module 6 App CPU 2, SW: Version 07.2.00T71
Compiled on Sep 25 2000 at 21:33:50 labeled as wsm-cpu3b
DRAM 134M, BRAM 262K, FPGA Version 0050
Code Flash 4M: Primary (880346 bytes, 07.2.00T71),
                Secondary (871842 bytes, 07.0.00T71)
Boot Flash 131K, Boot Version 06.00.00
The system uptime is 0 day 1 hour 54 minute 17 second
General Status: 0 ipc msg rec, 2 ipc msg sent
-----
Module 6 App CPU 3, SW: Version 07.2.00T71
Compiled on Sep 25 2000 at 21:33:50 labeled as wsm-cpu3b
DRAM 268M, BRAM 262K, FPGA Version 0050
Code Flash 4M: Primary (880346 bytes, 07.2.00T71),
                Secondary (871842 bytes, 07.0.00T71)
Boot Flash 131K, Boot Version 06.00.00
The system uptime is 0 day 1 hour 54 minute 17 second
General Status: 0 ipc msg rec, 2 ipc msg sent
```

**Syntax:** show wsm-state

This command displays the state of the modules in the chassis, the software version running on the modules, and detailed information for each processor on the modules.

### USING THE WEB MANAGEMENT INTERFACE

You cannot display general Web Switching Management Module information using the Web management interface.

## Determining Module Status

You can determine the status of a Web Switching Management Module in the following ways:

- Status LEDs – Each WSM CPU has LEDs that show send and receive activity for the processor. The MP has LEDs for data activity (both send and receive) and power.
- Module information in software – The module information displayed by the software indicates whether the module came up properly.

## Status LEDs

You can determine the status of a Web Switching Management Module processor by observing its LEDs. The processors have the following LEDs. Each WSM CPU has its own column of TxAct and RxAct LEDs. The left column shows activity for WSM CPU 1, the middle column shows activity for WSM CPU 2, and the right column shows activity for WSM CPU 3.

**Table 4.5: Web Switching Management Module LEDs**

LED	Position	State	Meaning
Active	Upper LED to the left of the serial interface	On	The MP is active.
		Off	The MP is not active.
Power	Lower LED to the left of the serial interface	On	The power status is good.
		Off	The power status is not good.
TxAct	Upper LED near the middle of the module	Blinking	The WSM CPU is transmitting data.
RxAct	Lower LED near the middle of the module	Blinking	The WSM CPU is receiving data.

## Software

You can display status information for a Web Switching Management Module using either of the following methods.

---

### NOTE:

- Slots on a four-slot chassis are numbered 1 – 4, from top to bottom.
  - Slots on an eight-slot chassis are numbered 1 – 8, from left to right.
- 

### USING THE CLI

To display the status of a Web Switching Management Module using the CLI, enter the following command at any CLI level:

```
ServerIron(config)# show module
Module                               Status    Ports Starting MAC
S1:
S2: Configured as B0GMR WSM Management Module
S3: B24E Copper Switch Module        OK        24    00e0.52c2.9f40
S4: B24E Copper Switch Module        OK        24    00e0.52c2.9f60
S5:
S6: B0GMR WSM Management Module      WSM, ACTIV 0
S7:
S8:
```

### Syntax: show module

The Status column shows the module status. A Web Switching Management Module can have one of the following statuses:

- ACTIVE – The module is currently the active management module.
- STANDBY – The module is the standby management module. (This applies only to management modules that support redundancy.)
- COMING UP – The module is coming up as the standby module. This status can be observed during switchover.
- FAILED – This status indicates that the host module failed to come up.
- OK – This status indicates that the module came up and is operating normally.

---

**NOTE:** The ACTIVE, STANDBY, and COMING UP status values apply only to management modules.

---

### USING THE WEB MANAGEMENT INTERFACE

1. Select the [System](#) link to display the System configuration sheet, if not already displayed.
2. Select the [Module](#) link to display the Module panel. The Status column shows the module status. A Web Switching module can have one of the following statuses:
  - ACTIVE – The module is currently the active management module.
  - STANDBY – The module is the standby management module. (This applies only to management modules that support redundancy.)
  - COMING UP – The module is coming up as the standby module. This status can be observed during switchover.
  - FAILED – This status indicates that the host module failed to come up.

- OK – This status indicates that the module came up and is operating normally.

---

**NOTE:** The ACTIVE, STANDBY, and COMING UP status values apply only to management modules.

---

### **Displaying Status from the Remote Console**

To display Web Switching Management Module status information while logged in to a WSM CPU (remote login), enter the following command from a remote login prompt:

```
ServerIron2/1 # wsm common show slot
slot 6: WSM Module with management cpu enabled
WSM1: alive, WSM2: alive, WSM3: alive,
```

This command shows brief processor information for the Web Switching Management Module. In this example, the module is a management module ("management cpu enabled") and the three WSM CPUs are operating normally.

### **Determining the Slot Allocations for the WSM CPUs**

The Web Switching Management Module automatically load balances Layer 4 – 7 processing by allocating chassis slots to the WSM CPUs according to the total bandwidth of the modules in the slots. To list the slot allocations, use the following CLI method.

#### **USING THE CLI**

To display the slot allocations for the WSM CPUs, enter the following command at any CLI level:

```
ServerIron(config)# show wsm-map
slot 2 (weight 24 x 100M) is processed by WSM 1/2 (weight 24)
slot 3 (weight 8 x 1000M) is processed by WSM 1/1 (weight 80)
slot 4 (weight 24 x 100M) is processed by WSM 1/3 (weight 24)
```

#### **Syntax:** show wsm-map

This example shows the slot allocations for a four-slot chassis. The output displays rows only for the slots that contain forwarding modules. No information is displayed for empty slots.

Each row shows the following information:

- The chassis slot ("slot 2" in the first row of the example above)
- The weight of the module in the slot ("weight 24 x 100M" in the first row of the example above)
- The chassis slot that contains the Web Switching Management Module and the WSM CPU to which the forwarding module described by this row is allocated ("is processed by WSM 1/2"). The "1" in this example indicates the Web Switching Management Module is in chassis slot 1. The "2" in this example indicates that WSM CPU 2 is handling Layer 4 – 7 processing for the forwarding module in slot 2.
- The total weight assigned to the WSM CPU ("weight 24" in the first row of this example)

---

**NOTE:** If the ports on a module are not up, the output says "will be processed" instead of "is processed" and the weight is listed as "0". In this case, the Web Switching Management Module reserves a WSM CPU for the module but does not add weight for the module's ports to the reserved WSM CPU.

---

---

**NOTE:** For reference, this example matches "Example Configuration 1" on page 4-3.

---

## Changing Slot Allocations for the WSM CPUs

---

**NOTE:** Foundry recommends that you change slot allocations only if Foundry technical support advises the change or the documentation for a feature states that the change is required.

---

To change slot allocations from the ones assigned by the device itself, enter a command such as the following at the global CONFIG level of the CLI:

```
ServerIron(config)# wsm wsm-map slot 3 wsm-slot 2 wsm-cpu 1
```

This command remaps processing for the forwarding module in slot 3 to WSM CPU 1 on the Web Switching Management Module in slot 2.

**Syntax:** wsm wsm-map <from-slotnum> wsm-slot <to-slotnum> wsm-cpu <cpunum>

The <from-slotnum> parameter specifies the slot that contains the module that you are mapping to another module for processing.

The <to-slotnum> parameter specifies the slot that contains the WSM CPU that will perform the processing.

The <cpunum> parameter specifies the WSM CPU on <to-slotnum> that will perform the processing. The WSM CPUs are numbered from 1 – 3.

## Additional Display Commands

The Web Switching Management Module has additional commands used by Foundry for troubleshooting:

- **wsm common show ipc** – Displays statistics for communication among the processors on the module.
- **wsm debug server wsm-cpu <slotnum> cpu** – Enables the debug mode.

---

**NOTE:** You must be logged in to a WSM CPU (remote login) to enter these commands. See “Changing the Management Session from the MP to a WSM CPU” on page 4-9.

---

---

# Chapter 5

## Configuring Hot Standby Redundancy

This chapter describes how to configure two ServerIrons for hot standby redundancy. Hot standby redundancy is a backup feature that allows a Foundry ServerIron to serve as an automatic backup for another ServerIron.

When ServerIrons are configured as backups, one ServerIron serves as the primary or active ServerIron, and the other serves as the secondary or standby ServerIron. The standby ServerIron becomes active only if the primary ServerIron fails due to loss of power or loss of data path. The active and standby ServerIrons must have the same SLB or TCS configuration and share the same virtual MAC address. The shared MAC address can be selected from the available pool on either the active ServerIron or the standby ServerIron.

Continuous communication of Layer 2 data, Layer 4 data, and power status is sent to the standby ServerIron by the active ServerIron through a dedicated private link. The private link can be a single port or can be multiple ports configured as a trunk group. To assign private links, you designate the ports that connect the two ServerIrons together as “backup” ports.

Each hot-standby pair consists of two ServerIrons. You can configure more than one hot-standby pair in the same broadcast domain if needed. See “Configuring a Backup Group ID” on page 5-4.

---

**NOTE:** As an alternative to hot standby, you can configure Symmetric Server Load Balancing (SLB). Symmetric SLB provides redundancy for individual virtual IP addresses (VIPs) while actively using all the ServerIrons. You do not need to dedicate a ServerIron as a standby. See “Configuring Symmetric SLB and SwitchBack” on page 7-1.

---

---

**NOTE:** This chapter does not apply to active-standby (“IronClad” or “high-availability”) configurations in Firewall Load Balancing (FWLB). For information about configuring FWLB redundancy, see the *Foundry ServerIron Firewall Load Balancing Guide*.

---

---

**NOTE:** The Spanning Tree Protocol (STP) interferes with the hot standby communication between the two ServerIrons. Although some backup topologies can appear to result in a logical loop, the backup ServerIron does not forward traffic. Therefore, no loop occurs. See Figure 5.2 for an example.

---

### How Hot Standby Redundancy Works

When a ServerIron configured for redundancy boots up, the device comes up as a backup system by default and checks to see if a private link is present.

- If a private link is not present, the ServerIron becomes the active partner in the pair.
- If a private link is present, a random number listening-time is initiated. The ServerIron listens for the presence of a primary ServerIron through the backup monitoring port.

- If the ServerIron detects a primary through its backup monitoring port, the ServerIron is placed in standby mode.
- If the ServerIron does not detect a primary within one second and the link status is good, then the ServerIron becomes the primary ServerIron when the listening-time expires.

To ensure against both ServerIrons becoming active at the same time, in the event that the dedicated link between the active and backup ServerIrons becomes unavailable, the active ServerIron also checks through the Layer 2 network for the presence of another active ServerIron.

Failover occurs if the active ServerIron becomes unavailable, has fewer available router ports (ports connected to routers) than the standby ServerIron, or loses connection to the servers.

### Failover In Software Release 07.3.03 and Later

In software release 07.3.03, the hot standby mechanism is enhanced to cause failover even if the active and standby ServerIrons have the same number of healthy router ports, when the active ServerIron has fewer healthy server ports. In previous releases, failover in this situation occurs only if the active ServerIron has zero healthy server ports.

Here is the algorithm for hot standby in software release 07.3.03:

1. Does the active ServerIron have more healthy router ports than the standby ServerIron?
  - Yes – The active ServerIron remains active.
  - No – Go to Step 2.
2. Do the active and standby ServerIrons have the same number of healthy router ports?
  - Yes – Go to Step 3.
  - No – SLB fails over to the standby ServerIron since it has more healthy router ports.
3. Does the active ServerIron have the same number or more of healthy server ports than the standby ServerIron?
  - Yes – The active ServerIron remains active.
  - No – SLB fails over to the standby ServerIron since it has more healthy server ports, even though both ServerIrons have the same number of healthy router ports.

The algorithm is not configurable. A configuration saved using an earlier release will use the new algorithm once you upgrade the software.

## Configuring Two ServerIrons for Hot Standby

In a typical hot standby configuration, one ServerIron is the active device and performs all the Layer 2 switching as well as the Layer 4 SLB switching while the other ServerIron monitors the switching activities and remains in a hot standby role. If the active ServerIron becomes unavailable, the standby ServerIron immediately assumes the unavailable ServerIron's responsibilities. The failover from the unavailable ServerIron to the standby ServerIron happens transparently to users.

Both ServerIrons share a common MAC address known to the clients. Therefore, if a failover occurs, the clients still know the ServerIron by the same MAC address. The active sessions running on the clients continue and the clients and routers do not need to re-ARP for the ServerIron MAC address.

To configure a pair of ServerIrons for hot standby redundancy, use one of the following methods.



**NOTE:** You can minimize your administrative tasks by completely configuring one of the ServerIrons, saving the configuration file, then copying the configuration file onto the other ServerIron. Just change the management IP address and source IP addresses on the second ServerIron and save the configuration file. The second ServerIron now contains all the feature configuration information that the first ServerIron contains.

If you plan to configure real servers to use a source IP address configured on the ServerIron as a default gateway, use the **source-standby-address** or **source-nat-address** command rather than the **source-ip** or **source-nat** command.

---

**NOTE:** These procedures include steps to save the configuration changes to the startup-config file and then reload the software. Make sure you perform these steps.

---

### *USING THE CLI*

To configure port 13 as the hot standby port on the first ServerIron, enter the following commands:

```
ServerIron1(config)# vlan 2
ServerIron1(config-vlan-2)# untag ethernet 13
ServerIron1(config-vlan-2)# no spanning-tree
ServerIron1(config-vlan-2)# exit
ServerIron1(config)# server backup ethernet 13 00e0.5201.0c72
ServerIron1(config)# write memory
ServerIron1(config)# end
ServerIron1# reload
```

The first three commands place the hot standby port in its own port-based VLAN and disable STP on the VLAN. Placing the hot standby port in its own VLAN protects the port from broadcast storms or other network issues on the other ServerIron ports. The exit commands returns the CLI to the global CONFIG level.

The **server backup** command designates port 13 as the hot standby port and specifies the MAC address the port is backing up. The **write memory** command saves the configuration changes to the device's startup-config file. The **end** command returns the CLI to Privileged EXEC level. You must be at this level to enter the **reload** command, which reloads the software and places the hot standby configuration into effect.

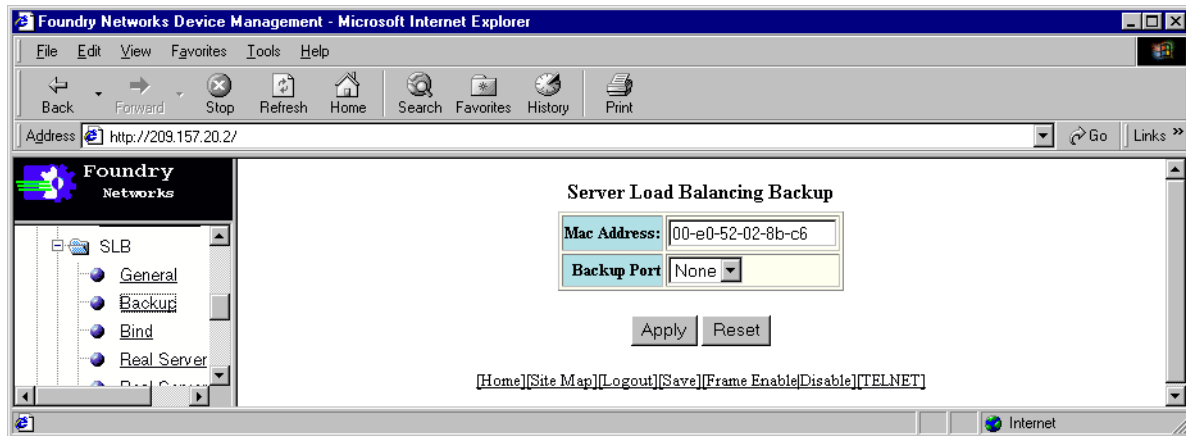
To configure port 4 as the hot standby port on the second ServerIron, enter the following commands. Notice that the hot standby MAC address on each ServerIron is the same. Using the same MAC address is a requirement.

```
ServerIron2(config)# vlan 2
ServerIron2(config-vlan-2)# untag ethernet 13
ServerIron2(config-vlan-2)# no spanning-tree
ServerIron2(config-vlan-2)# exit
ServerIron2(config)# server backup ethernet 4 00e0.5201.0c72
ServerIron2(config)# write memory
ServerIron2(config)# end
ServerIron2# reload
```

### *USING THE MANAGEMENT INTERFACE*

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the Backup link to display a panel such as the one shown in Figure 5.1

**Figure 5.1 SLB Backup panel**



5. Enter the MAC address that the two ServerIrons will share.
6. Select the port that connects this ServerIron to the other ServerIron.
7. Click the Apply button to apply the change to the device's running-config file.
8. Click on the plus sign next to VLAN in the tree view of Configure options.
9. Select the Port link to display the Port VLAN table. This table lists the port-based VLANs configured on the device. By default, VLAN 1 is already configured and all ports are members of the VLAN.
10. Select the Add Port VLAN link under the table.
11. Optionally change the VLAN ID in the VLAN ID field. The Web management interface automatically increments the value in the field to the next available VLAN ID.
12. Optionally enter a name for the VLAN.
13. Optionally change the QoS level.
14. Select Disable next to Spanning Tree. You must disable STP on the VLAN, since you will be using the port(s) you place in the VLAN as the hot standby port(s).
15. Select the port(s) in the Port Members section. In this example, select port 13 for ServerIron1 or port 4 for ServerIron2.
16. Select Add to add the port(s) to the VLAN. The software automatically removes the port(s) you add to this VLAN from VLAN 1 (the default VLAN, which contains all the device's ports).
17. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.
18. Click on the plus sign next to Command in the tree view to list the command options.
19. Select the Reload link and select Yes when the Web management interface asks you whether you really want to reload the software.
20. Repeat Step 1 – Step 19 on the other ServerIron.

## Configuring a Backup Group ID

You can configure up to eight hot-standby pairs within a single broadcast domain. To enable this support, you configure a backup group ID on each of the ServerIrons, so that both ServerIrons in a given pair have the same ID. The backup group ID uniquely identifies the pair.

When you configure a backup group ID, both ServerIrons in a hot-standby pair use the ID when exchanging backup information. If a ServerIron receives a backup information packet but the packet's backup group ID does not match the ServerIron's backup group ID, the ServerIron discards the packet.

If the broadcast domain contains multiple hot-standby pairs, you must configure backup group IDs on all pairs. If the broadcast domain contains only one hot-standby pair, you do not need to configure a backup group ID.

To configure a backup group ID, enter a command such as the following at the global CONFIG level of the CLI:

```
ServerIron(config)# server backup-group 1
```

**Syntax:** [no] server backup-group <num>

The <num> parameter specifies the backup group ID and can be a number from 0 – 7. Enter the same ID on both ServerIrons in a hot-standby pair. Do not enter the same ID on a ServerIron that is not one of the ServerIrons in the hot-standby pair.

## Changing the Backup Timer

The standby ServerIron assumes the active role if the standby ServerIron does not receive a Hello message or Layer 4 session synchronization data from the active ServerIron within a certain number of seconds since receiving the last Hello message or synchronization data.

By default, the standby ServerIron waits one second since receiving the last Hello message or data to receive a new message or data. If the standby ServerIron does not receive a new Hello message or data within one second, the standby ServerIron assumes that the active ServerIron is no longer available and takes over the active role.

In some configurations, particularly configurations in which the active ServerIron is performing a lot of processing, it is possible for frequent failovers to occur. In this situation, although the active ServerIron is still available and actively serving load balancing or other requests, the active ServerIron does not always send the Hello message or synchronization data in time for the standby ServerIron. As a result, the standby ServerIron takes over the active role. If similar conditions cause the newly active ServerIron to sometimes miss sending the Hello messages or synchronization data in time, failover occurs again.

You can prevent unnecessary state flapping between the two ServerIrons by increasing the backup timer. When you increase the backup timer, the standby ServerIron waits longer to receive new Hello messages or synchronization data from the active ServerIron. As a result, flapping is reduced or eliminated.

To change the backup timer, use one of the following methods.

---

**NOTE:** The backup timer must have the same value on both ServerIrons in the active-standby pair.

---

### USING THE CLI

To change the backup timer on a ServerIron in an active-standby pair, enter a command such as the following at the global CONFIG level of the CLI:

```
ServerIron(config)# server backup-timer 50
```

This command sets the backup timer to 5 seconds (50 \* 100 milliseconds).

**Syntax:** server backup-timer <time>

The <time> parameter specifies how long this ServerIron, when it is the backup ServerIron, will wait for a Hello message or synchronization data from the active ServerIron before assuming the active ServerIron is no longer available. You can specify a value from 5 (one half second) – 100 (10 seconds), in units of 100 milliseconds each. The default is 10 (one second).

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu.
5. Enter a value from 5 (one half second) – 100 (10 seconds), in units of 100 milliseconds each, in the Backup timer field. The default is 10 (one second).

6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Configuring a ServerIron to Always be the Active Partner

You can configure one of the ServerIrons in the active-standby pair to always be the active ServerIron. When you enable this option on one of the ServerIrons, that ServerIron is always the active one by default. The only event that can cause the other ServerIron to be the active one is unavailability of the default active ServerIron or its link to the backup ServerIron. To allow graceful insertion, the ServerIron does not immediately assume the active role, but instead waits for a configurable number of minutes before taking the active role.

To configure a ServerIron in an active-standby pair to always be the active ServerIron, use one of the following methods.

#### USING THE CLI

To configure a ServerIron in an active-standby pair to always be the active ServerIron, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# server backup-preference 5
```

**Syntax:** server backup-preference <wait-time>

The <wait-time> parameter specifies how long the ServerIron waits before assuming the active role. The ServerIron does not immediately become the active ServerIron but instead waits the number of minutes you specify. You can specify from 5 – 30 minutes. This parameter does not have a default.

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu.
5. Enter a value from 5 – 30 in the Backup preference field to specify the number of minutes you want the ServerIron to wait before assuming the active role. This parameter does not have a default.
6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Viewing Hot Standby Information

After you configure a ServerIron for hot standby redundancy, you can verify the configuration using the following methods.

#### USING THE CLI

To view the hot standby configuration, enter the following command:

```
ServerIron1(config)# show server backup
```

#### USING THE MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the Backup link.

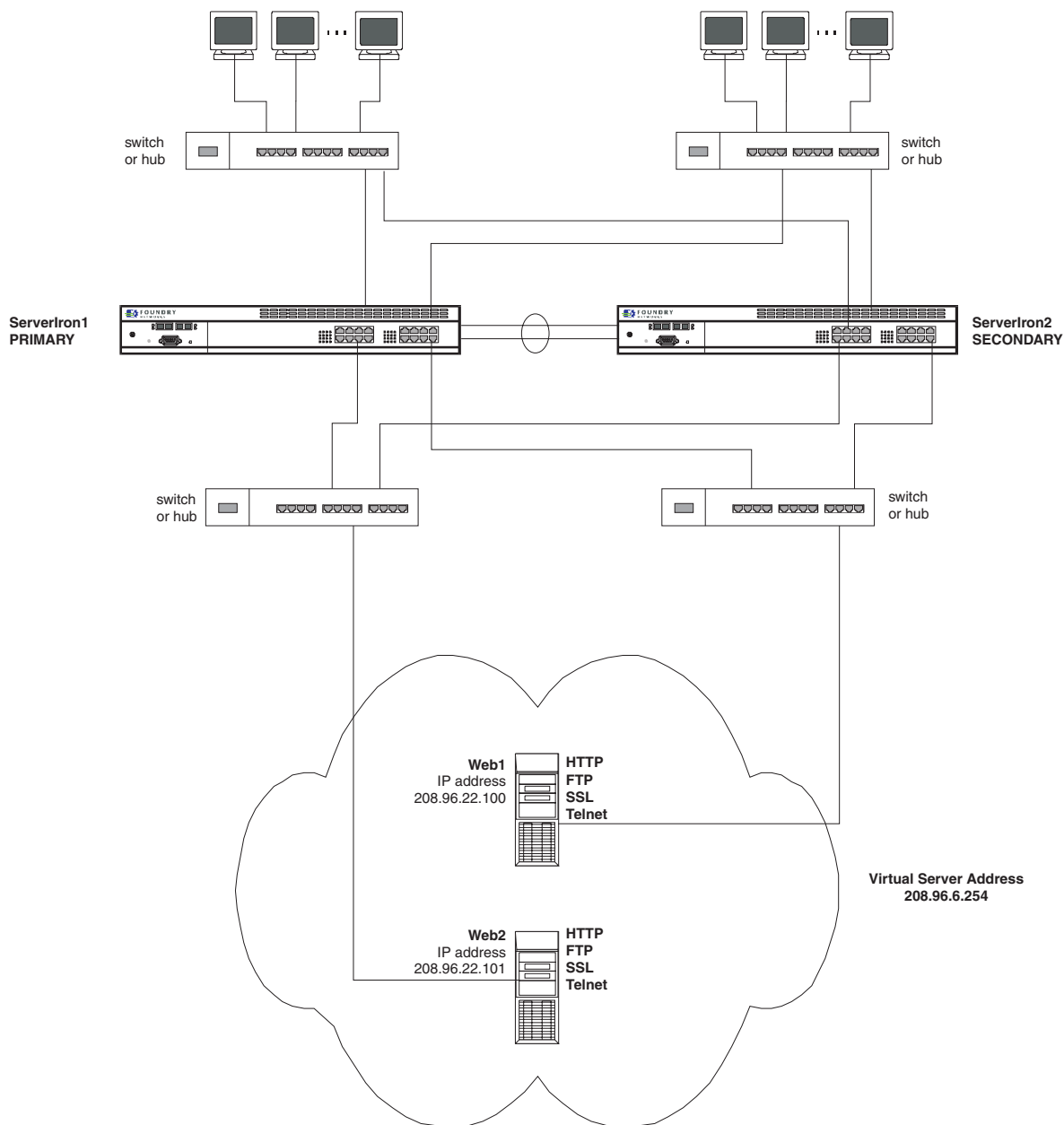
## Hot Standby Configuration Example

Hot standby redundancy allows a Foundry ServerIron to serve as automatic backup to another switch.

Suppose you want to configure a second switch, ServerIron2, to serve as the backup or standby switch for ServerIron1. Each switch will be configured with the same SLB configuration, supporting the following TCP/UDP ports: HTTP, SSL, FTP, and Telnet.

The private link, which provides the connection between the active and standby switches, will be configured as a trunk group with ports 13 and 14 as members.

**Figure 5.2 ServerIron2 serving as standby switch to ServerIron1**



To configure the setup shown in Figure 5.2, perform the following tasks:

1. Configure a trunk group.

---

**NOTE:** When you configure the trunk group, you must use the "server" parameter instead of the default "switch" parameter, regardless of the device type you are connecting to.

---

2. Define the real server and service.
3. Define the virtual server and service.
4. Bind the virtual and real services.
5. Configure a backup port for redundancy. Make sure you place the backup port in its own port-based VLAN and disable STP on the VLAN.

---

**NOTE:** Although the topology shown in Figure 5.2 appears to result in a logical loop, the backup ServerIron does not forward traffic. Therefore, no loop occurs. However, STP will interfere with the hot standby communication between the two ServerIrons if you leave STP enabled on the VLANs that contain the hot standby ports.

---

### USING THE CLI

To configure the example in Figure 5.2, enter the following commands:

```
ServerIron# config term
ServerIron(config)# trunk server ethernet 13 to 14
```

---

**NOTE:** On all models **except** the ServerIron 400 and ServerIron 800, you must use the "server" parameter instead of the default "switch" parameter, regardless of the device type you are connecting to. This applies even if the other device is a ServerIron.

---

On the ServerIron 400 or ServerIron 800, you must use the default trunk type, "switch" (example: **trunk switch ethernet 1/1 to 1/2**).

---

```
ServerIron(config)# vlan 2
ServerIron(config-vlan-2)# untag ethernet 13 to 14
ServerIron(config-vlan-2)# no spanning-tree
ServerIron(config-vlan-2)# exit
ServerIron(config)# server real web1 208.96.22.100
ServerIron(config-rs-web1)# port http
ServerIron(config-rs-web1)# port ssl
ServerIron(config-rs-web1)# port ftp
ServerIron(config-rs-web1)# port telnet
ServerIron(config-rs-web1)# server real web2 208.96.22.101
ServerIron(config-rs-web2)# port http
ServerIron(config-rs-web2)# port ssl
ServerIron(config-rs-web2)# port ftp
ServerIron(config-rs-web2)# port telnet
ServerIron(config-rs-web2)# server virtual www.alterego.com 208.96.6.254
ServerIron(config-vs-www.alterego.com)# port http
ServerIron(config-vs-www.alterego.com)# port ssl sticky
ServerIron(config-vs-www.alterego.com)# port ftp
ServerIron(config-vs-www.alterego.com)# port telnet
ServerIron(config-vs-www.alterego.com)# bind http web1 http web2 http
ServerIron(config-vs-www.alterego.com)# bind ssl web1 ssl web2 ssl
ServerIron(config-vs-www.alterego.com)# bind ftp web1 ftp web2 ftp
ServerIron(config-vs-www.alterego.com)# bind telnet web1 telnet web2 telnet
ServerIron(config-vs-www.alterego.com)# exit
```

To configure the trunk group, assign ports 13 and 14 as the backup ports, assign round robin as the predictor (load balancing metric), and disable Spanning Tree, enter the following commands:

```
ServerIron(config)# server backup ethernet 13 00e0.5201.0c72
```

```
ServerIron(config)# server predictor round-robin
ServerIron(config)# no span
ServerIron(config)# exit
ServerIron# write memory
ServerIron# reload
```

The MAC address assigned is a MAC address that is resident on either ServerIron1 or ServerIron2. Notice that because port 13 is the lead port for the trunk group, you do not need to configure any other ports within that group.

---

**NOTE:** A backup port should be configured on both switches to form the private link. Additionally, the private link should be configured on both switches before bringing the standby switch into the network or cabling the two units together through the private link.

---

### *USING THE MANAGEMENT INTERFACE*

---

**NOTE:** When configuring the trunk group, you must use the “server” parameter instead of the default “switch” parameter, regardless of the device type you are connecting to. This applies even if the other device is a ServerIron.

---

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Select the Trunk link.
4. Select the box next to the ports. For example, select the box next to the “port 13 - 16” option and select 2 as the number of ports to assign ports 13 and 14 to a trunk group.
5. Click the Apply button to apply the change to the device’s running-config file.
6. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device’s flash memory.

### **Defining the Real Server**

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the Real Server link.

---

**NOTE:** If real servers are already defined on the switch, select the Add Real Server link to reach the Real Server entry panel.

---

5. Enter the real server name. In this case, enter either “web1” or “web2”.
6. Enter the IP address of the web server.
7. Modify the weight connections or weight fields, if desired. In this case, no change is made to the system defaults.
8. Select the Add button to assign the changes.
9. Click the Apply button to apply the change to the device’s running-config file.
10. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device’s flash memory.

---

**NOTE:** Repeat steps 5 – 10 for real servers web1 and web2.

---

### Defining the Real Server Ports

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the Real Server Port link.

---

**NOTE:** If real server ports are already defined, select Add Real Server Port.

---

5. Select the server name from the pulldown menu. In this case, select either "web1" or "web2".
6. Select the TCP/UDP port to be assigned to the server. In this case, assign ports HTTP, SSL, FTP, and Telnet to both web1 and web2.
7. Select the Add button after each port selection.
8. Select the Enable option after you add all the ports.
9. Select the Add button to assign the changes.
10. Select Show Real Server Port to display the configuration results.
11. Click the Apply button to apply the change to the device's running-config file.
12. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Defining Virtual Server and Service

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the Virtual Server link.

---

**NOTE:** If any virtual servers are already defined on the switch, a summary panel appears instead. Select the Add Virtual Server Port link.

---

5. Enter the server name of the virtual server.
6. Enter the IP address of the virtual server in the Server IP field.
7. Select the load balancing metric to be used by the virtual server.

---

**NOTE:** A global default exists for all servers. You can modify this on a server-by-server basis. If you want to operate with the global metric defined on the SLB configuration sheet, then select the Default button. To assign a metric other than the global default for a specific server, select one of the other options not already defined as the global default.

---

8. Select the Add button to assign the configured virtual server.
9. Click the Apply button to apply the change to the device's running-config file.
10. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Defining Virtual Server Ports

To assign the TCP/UDP service ports to be supported by the virtual server:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.



3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the Virtual Server Port link.
5. Select the virtual server name to which a service is to be assigned from the Server Name pulldown menu.
6. Select the ports to be supported from the TCP/UDP Port pulldown menu. In this case, enter Telnet, FTP, HTTP, and SSL separately.
7. Enable the ports.
8. Enable the Sticky option if your users will need to reconnect to the same server for the ports. For this example, enable the sticky port option when you add the SSL port.
9. Enable the concurrent option if you want all secondary connections to attach to the same server to which the first connection attaches.
10. Select the Add button to assign the TCP/UDP ports to the virtual server.
11. Click the Apply button to apply the change to the device's running-config file.
12. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Binding Real and Virtual Server Ports

After you define the virtual and real servers and the supported ports, you must bind the appropriate virtual and real servers to one another. Once this is done, all traffic received for the virtual address is forwarded to the defined real server(s).

To bind a virtual and real server:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the Bind link.
5. Select the virtual server name from the pulldown menu.
6. Select the virtual server TCP/UDP port from the pulldown menu.
7. Select the real server name from the pulldown menu.
8. Select the real TCP/UDP port from the pulldown menu.
9. Select the Bind button to map the selected virtual server and service to the selected real server and server port.

### Configuring the Backup Port for SLB Redundancy

After you define the virtual server and real server and their service and they are bound, you can assign a backup port to support the redundancy link between the two servers.

To configure a port as a backup port:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the Backup link to display a panel such as the one shown in Figure 5.1
5. Enter the MAC address that the two ServerIrons will share.
6. Select the port that connects this ServerIron to the other ServerIron.
7. Click the Apply button to apply the change to the device's running-config file.

8. Click on the plus sign next to VLAN in the tree view of Configure options.
9. Select the [Port](#) link to display the Port VLAN table. This table lists the port-based VLANs configured on the device. By default, VLAN 1 is already configured and all ports are members of the VLAN.
10. Select the [Add Port VLAN](#) link under the table.
11. Optionally change the VLAN ID in the VLAN ID field. The Web management interface automatically increments the value in the field to the next available VLAN ID.
12. Optionally enter a name for the VLAN.
13. Optionally change the QoS level.
14. Select Disable next to Spanning Tree. You must disable STP on the VLAN, since you will be using the port(s) you place in the VLAN as the hot standby port(s).
15. Select the port(s) in the Port Members section.
16. Select Add to add the port(s) to the VLAN. The software automatically removes the port(s) you add to this VLAN from VLAN 1 (the default VLAN, which contains all the device's ports).
17. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.
18. Click on the plus sign next to Command in the tree view to list the command options.
19. Select the [Reload](#) link and select Yes when the Web management interface asks you whether you really want to reload the software.
20. Repeat Step 1 – Step 19 on the other ServerIron.

## Displaying Hot Standby Configuration Information

After you configure ServerIrons for hot standby, you can view the backup configuration information. To do so, use one of the following methods.

### USING THE CLI

To display backup configuration information, enter the following command at any level of the CLI:

```
ServerIron> show server backup
Server Backup port configured

Switch state = Active
SLB state    = 0
SLB Partner MAC valid= 0
SLB Partner MAC      = abcd.abcd.abcd
SLB Partner port cnt = 24
Transitions, activates =          0,          standby =          0
Pdus sent           =          10, Mac pdu sent =          10
No pdus              =          0, no port maps =          0
```

The first row in the display indicates whether the ServerIron is configured for hot standby. The following table describes the remaining fields in this display.

---

**NOTE:** Some field names refer to SLB but these fields also apply to TCS and to firewall load balancing.

---

**Table 5.1: Hot Standby Information**

This Field...	Displays...
Switch state	Indicates whether this ServerIron is the active ServerIron or the standby. The state can be one of the following: <ul style="list-style-type: none"> <li>• Active</li> <li>• Standby</li> </ul>
SLB state	This field should contain a 0 on both the active and standby ServerIrons.
SLB Partner MAC valid	Indicates whether the SLB partner MAC address listed in the SLB Partner MAC field is valid. The value can be one of the following: <ul style="list-style-type: none"> <li>• 0 – invalid</li> <li>• 1 – valid</li> </ul>
SLB Partner MAC	The MAC address of port 1 on the other ServerIron, thus indicating Layer 2 connectivity between the ServerIrons. If this field contains all zeros, double-check the connection between the ServerIrons and verify that both ServerIrons are powered on. Also verify that Spanning Tree is disabled on both ServerIrons. Spanning Tree interferes with Hot Standby.
SLB Partner port cnt	The number of physical ports on the other ServerIron.
Transitions, activates	The number of times this ServerIron has changed from standby to active.
Transitions, standby	The number of times this ServerIron has changed from active to standby.
Pdus sent	The number of Layer 4 synchronization packets this ServerIron has sent to the other ServerIron.
Mac pdu sent	The number of MAC-layer synchronization packets this ServerIron has sent to the other ServerIron.
No pdus	The number of missed Layer 4 or MAC-layer PDUs.
no port maps	The number of missed port map PDUs. Port map PDUs are used by the ServerIron to discover information about the maps on the other ServerIron.

### **USING THE WEB MANAGEMENT INTERFACE**

To display the number of the backup port on a ServerIron:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitor options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.

4. Select the General link. The port number is listed in the Backup Port field. If the field contains the value "None", then hot standby is not configured.

---

## Chapter 6

# Configuring Server Load Balancing

This chapter describes how to configure Server Load Balancing (SLB) on the ServerIron using the Command Line Interface (CLI) and Web management interface. See the *Foundry ServerIron Command Line Interface Reference* for information about the CLI commands.

To display SLB configuration information and statistics, see “Viewing SLB Configuration Details and Statistics” on page 6-69.

Application examples at the end of chapter show practical applications of the SLB features. See “SLB Application Examples” on page 6-96.

SLB is based on associations between real servers and virtual servers. The real servers are your application servers. The virtual servers have one or more virtual IP addresses (VIPs). You associate a real server with a virtual server by binding TCP/UDP ports on the real servers with TCP/UDP ports on the virtual server. When a client sends a TCP/UDP request for a port on the virtual server, the ServerIron sends the client's request to the real server.

The client is unaware of the real servers behind the virtual server but does experience enhanced throughput and availability for TCP/UDP services.

## SLB Parameters

Table 6.1 lists the SLB parameters and where you can find more information about them in this document.

---

**NOTE:** The CLI shows some parameters that are not listed here. For example, real server ports include parameters for configuring slow-start and history groups. These parameters do not apply specifically to SLB and are described in other places in this guide.

---

**Table 6.1: SLB Parameters**

Parameter	Description	Default	See page...
<b>Global SLB Parameters</b>			
Fast-path SLB	An optimization option that uses fast-path processing. When you enable fast-path processing, the ServerIron does not process every TCP or UDP packet in a given session in detail. Instead, the ServerIron uses information gathered during setup of the session to forward packets in the session.  You can enable fast-path processing for either stateful SLB or stateless SLB.	Disabled	6-22

Table 6.1: SLB Parameters (Continued)

Parameter	Description	Default	See page...
Load Balancing Method (predictor)	<p>The method the ServerIron uses to select a real server for a client request. The following methods are supported:</p> <ul style="list-style-type: none"> <li>Least connections – The ServerIron sends the request to the real server that currently has the fewest active connections with clients.</li> <li>Least sessions – The ServerIron sends the request to the real server that currently has the fewest session table entries.</li> <li>Round-robin – The ServerIron sends the request to each server in rotation, regardless of how many connections or sessions each server has.</li> <li>Weighted – The ServerIron uses the weights you assign to the real servers to select a real server. The weights are based on the number of session table entries the ServerIron has for each server.</li> <li>Response time only – The ServerIron selects the real server with the fastest response time.</li> <li>Least connection and server response time weights – The ServerIron compares a combination of a real server's least-connections weight and server response time weight to the same values for the other real servers.</li> <li>Least local connections (ServerIron 400 or ServerIron 800 only) – On an individual WSM CPU basis, the ServerIron selects the real server with the fewest active connections with clients. The predictor selects the real server that has the least number of connections created by the local WSM CPU. The local WSM CPU is the CPU that is managing the chassis slot connected to the real server.</li> <li>Least local sessions (ServerIron 400 or ServerIron 800 only) – On an individual WSM CPU basis, the ServerIron selects the server that has the fewest active session on the WSM CPU attached to the real server. The number of sessions is updated when session entries are deleted.</li> </ul>	Least connections	6-24
Router ports	The ServerIron Ethernet ports that are connected to routers.	None configured	6-27
TCP SYN limit	The maximum number of TCP SYN requests the ServerIron will send to a real server in a one-second interval. You can specify from 1 – 65535 requests a second.	65535	6-27

**Table 6.1: SLB Parameters (Continued)**

Parameter	Description	Default	See page...
Server response threshold	The maximum number of milliseconds a real server's reply can take to reach the ServerIron before the ServerIron generates a warning message or shuts down (stops using) the server. You can specify separate warning and shutdown thresholds.	None configured	6-28
Layer 3 health checks	Disables or re-enables the Layer 3 health check (IP ping) sent by the ServerIron to local real servers or remote real servers when you configure the servers.	Enabled	12-20
ICMP unreachable messages	If an application port requested by a client is not available, the ServerIron can send an ICMP Destination Unreachable message to the client.	Disabled	6-29
Sending TCP RST messages or ICMP Destination Unreachable messages to clients	You can configure the ServerIron to send a TCP RST or ICMP Destination Unreachable message to a client if the client's request for a TCP application on a VIP is not successful.	TCP RST: <ul style="list-style-type: none"> <li>• Enabled (07.2.25)</li> <li>• Disabled (other releases)</li> </ul> ICMP Unreachable: <ul style="list-style-type: none"> <li>• Disabled (all releases)</li> </ul>	6-30
Source IP address	A secondary IP address for the ServerIron. Source IP addresses are useful for placing the ServerIron in multiple IP sub-nets when the real servers and ServerIron are not in the same sub-net. You can configure up to eight source IP addresses.	None configured	6-30
Source NAT	Allows the ServerIron to use a source IP address as the source for packets sent to real servers.	Disabled	6-32
Reverse NAT	Allows the ServerIron to change the source IP address of traffic that the real server initiates on TCP or UDP ports that are bound to a VIP.	Disabled	6-33
Force shutdown	Shuts down an application within two minutes even if client connections are still open on the application. When this feature is disabled, the ServerIron allows all existing client connections on a service you have shut down to terminate normally.	Disabled	6-34



Table 6.1: SLB Parameters (Continued)

Parameter	Description	Default	See page...
Sticky age	Ages out inactive sticky server connections. A connection is sticky if you configure the ServerIron to send successive requests from the same client for the same application port to the same real server, instead of load balancing the requests to different real servers.  You can specify from 2 – 60 minutes.  To configure an application port to be sticky, see “Sticky” on page 6-61.	Five minutes  <b>Note:</b> This is the age timer default, but application ports are not sticky by default.	6-34
Persistent sticky connections	Continues using a sticky port even if you have entered a command to unbind the port or the port is disabled.	Disabled	6-35
Transparent VIP	Allows the ServerIron to load balance the same VIP with other ServerIrons, without “owning” the VIP.	Disabled	6-36
TCP fast aging	Following a RST from the server, ages out session table entries in the amount of time specified in the <b>server msl</b> command, by default 8 seconds.	Enabled	6-36
Decrementing the current connection counter immediately after server RST	You can configure the ServerIron to immediately decrement its current connection counter when it receives a RST from the server.	Disabled	6-36

**Application Port Parameters**

You can globally configure an application port by configuring a port profile for the port. To configure a port profile, see “Configuring a Port Profile” on page 12-21.

Port type (TCP or UDP)	Whether the application port is a TCP port or a UDP port.	UDP  <b>Note:</b> The ServerIron already knows the port type for the applications that are known to the ServerIron.	12-23
Keepalive state	Whether the periodic Layer 4 or Layer 7 health check for the application port is enabled.	Disabled	12-23
Keepalive interval and retries	How often the ServerIron performs a periodic health check for the application, and how many times the ServerIron retries an unsuccessful health check before changing the application’s state to SUSPECT.  You can specify from 2 – 120 seconds for the interval. You can specify from 1 – 5 retries.	<ul style="list-style-type: none"> <li>Interval – 2 seconds</li> <li>Retries – 5 seconds</li> </ul>	12-23
Keepalive port	An application port that you want the ServerIron to use for health checking another application port. For example, you can use an SSH port (443) to check the health of an HTTP port (80).	None configured	12-23

**Table 6.1: SLB Parameters (Continued)**

Parameter	Description	Default	See page...
TCP or UDP age	How long the ServerIron keeps an idle TCP or UDP session table entry before removing the entry from the table. You can set the TCP or UDP age from 2 – 60 minutes.	<ul style="list-style-type: none"> <li>TCP – 30 minutes</li> <li>UDP – 5 minutes</li> </ul>	12-23
Session synchronization	Whether the ServerIron provides failover for individual sessions in a Symmetric SLB configuration.	Disabled	7-1
Connection logging	Whether the ServerIron generates a Syslog entry when a session table entry is created for a port.	Disabled	12-60
Smooth factor	A value used by the server response time and weighted load balancing methods to smooth out multiple response time samples for a real server.	90	12-28
<b>Real Server Parameters</b>			
IP address	The IP address of the real server. <b>Note:</b> You specify the address when you configure the real server. To change the address after the server is configured, see “IP Address” on page 6-37.	None	6-37
Location	Whether the real server is local (connected to the ServerIron at Layer 2) or remote (connected through one or more router hops). The ServerIron load balances requests among local servers and uses remote servers only if all the local servers are unavailable.	None You specify whether the server is local or remote when you define the server.	6-37
Backup	Whether the server is a primary server or a backup server. <b>Note:</b> This parameter is configurable only on the ServerIron 400 or ServerIron 800.	Locally attached – primary Attached through a router – backup	6-38
Application ports	The TCP or UDP application ports for which you want the ServerIron to load balance requests.	None Specify the ports when you define the real server on the ServerIron.	6-40
Host ranges and host-range maps	Configures a contiguous range of IP addresses on the real server beginning with an IP address you specify. You must also configure a corresponding range of addresses on the virtual server.	None configured	6-41
Maximum connections	Limits the maximum number of sessions the ServerIron will maintain in its session table for a real server. You can specify from 1 – 1,000,000 sessions.	The maximum session table size, which is 1,000,000 by default	6-45
Traffic rate threshold	The maximum number of bytes per second allowed on the real server. The ServerIron uses the number of bytes in all received and transmitted TCP and UDP packets in its calculation of the traffic rate.	None configured	6-46

Table 6.1: SLB Parameters (Continued)

Parameter	Description	Default	See page...
Server response threshold	The maximum number of milliseconds a real server's replay can take to reach the ServerIron before the ServerIron generates a warning message or shuts down (stops using) the server. You can specify separate warning and shutdown thresholds.	None configured	6-46
Layer 3 health check	Disables or re-enables the Layer 3 health check (IP ping) sent by the ServerIron to a real server when you configure the server.	Enabled	6-47
Source NAT	Allows the ServerIron to use a source IP address as the source for packets sent to the real server. <b>Note:</b> You also can enable this feature globally for all real servers.	Disabled	6-47
Weight	The real server's weight. This parameter is used for the weighted and least connection with server response time weights load balancing methods.	Zero	6-48
<b>Real Server Application Port Parameters</b>			
Port type (TCP or UDP)	Whether the application port is a TCP port or a UDP port.  This parameter is not configurable at this level. The ServerIron already knows the port type for the application ports that are known to the ServerIron. For other application ports, the ServerIron assumes they are UDP unless you specify that the port is TCP. To specify an application port's type, configure a port profile for the port.	UDP  <b>Note:</b> The ServerIron already knows the port type for the applications that are known to the ServerIron.	12-23
Port state	Whether the port is enabled or disabled on the real server.	Enabled	6-50
Binding state	Whether the application ports are bound to virtual servers. You can globally unbind a real server's application ports from all virtual servers.	Bound to the virtual servers you bind them to.	6-50
Keepalive state	Whether continual Layer 4 or Layer 7 health checks are enabled.  Continual health checks are automatically enabled when you configure a port profile for a port.	Disabled  <b>Note:</b> If you have configured a port profile for the port, the keepalive health check is enabled.	6-50
Maximum new connection rate	Connection Rate Limiting (CRL) specifies the maximum number of new TCP, UDP, or individual port connections per second allowed on the real server.	None	6-51
Status code	Specifies the HTTP status codes that indicate a successful health check.  <b>Note:</b> This parameter applies only to HTTP ports.	200 – 299	12-32

**Table 6.1: SLB Parameters (Continued)**

Parameter	Description	Default	See page...
URL	The HTTP method (HEAD or GET) and the Universal Resource Locator (URL) the ServerIron requests when sending a health check to an HTTP port. <b>Note:</b> This parameter applies only to HTTP ports.	HEAD /1.0	12-32
Content match	Specifies web content on the real server that indicates a successful health check. <b>Note:</b> This parameter applies only to HTTP ports.	None configured	12-33
Address query	Specifies an address request for a specific domain name. If the server successfully responds with the IP address for the domain name, the server passes the health check. <b>Note:</b> This parameter applies only to DNS ports.	None configured	12-38
DNS zone	Specifies a Source-of-Authority (SOA) request for a specific zone name. If the server is authoritative for the zone and successfully responds to the SOA request, the server passes the health check. <b>Note:</b> This parameter applies only to DNS ports.	None configured	12-38
RADIUS username, password, and key	Specifies RADIUS parameters that the ServerIron sends to a RADIUS application port in the Layer 7 health check. <b>Note:</b> This parameter applies only to RADIUS ports.	None configured	12-38
LDAP version	Specifies the LDAP version to use in the LDAP Layer 7 health check. <b>Note:</b> This parameter applies only to LDAP ports.	None configured	12-39
<b>Virtual Server Parameters</b>			
Application ports and bindings	The TCP or UDP application ports for which you want the ServerIron to load balance requests. Add the ports to the virtual server, then bind the virtual server to the real server based on the ports.	None  Specify the ports when you define the virtual server.	6-52
Primary and backup servers	Configures the virtual server to use the primary and backup servers specifically designated to be primaries or backups, instead of using the locally attached servers as the primary load-balancing servers and the remotely attached servers as the backup servers. <b>Note:</b> This parameter is configurable only on the ServerIron 400 or ServerIron 800.	Locally attached servers – primary  Servers attached through a router – backup	6-53
Host ranges and host-range maps	Configures a range of IP addresses on the virtual server beginning with an IP address you specify. You must also configure a corresponding range of addresses on the real server.	None configured	6-54

**Table 6.1: SLB Parameters (Continued)**

Parameter	Description	Default	See page...
HTTP redirect	Configures the ServerIron to send an HTTP redirect message to the client so that the client redirects its HTTP connection to the real server's IP address instead of the VIP. Use this feature when you configure remote real servers and want replies from the servers to go directly to clients.	Disabled	6-54
Load balancing method	<p>The method the ServerIron uses to select a real server for a client request to this virtual server. The following methods are supported:</p> <ul style="list-style-type: none"> <li>• Least connections – The ServerIron sends the request to the real server that currently has the fewest active connections with clients.</li> <li>• Least sessions – The ServerIron sends the request to the real server that currently has the fewest session table entries.</li> <li>• Round-robin – The ServerIron sends the request to each server in rotation, regardless of how many connections or sessions each server has.</li> <li>• Weighted – The ServerIron uses the weights you assign to the real servers to select a real server. The weights are based on the number of session table entries the ServerIron has for each server.</li> <li>• Response time only – The ServerIron selects the real server with the fastest response time.</li> <li>• Least connection and server response time weights – The ServerIron compares a combination of a real server's least-connections weight and server response time weight to the same values for the other real servers.</li> <li>• Least local connections (ServerIron 400 or ServerIron 800 only) – On an individual WSM CPU basis, the ServerIron selects the real server with the fewest active connections with clients. The predictor selects the real server that has the least number of connections created by the local WSM CPU. The local WSM CPU is the CPU that is managing the chassis slot connected to the real server.</li> <li>• Least local sessions (ServerIron 400 or ServerIron 800 only) – On an individual WSM CPU basis, the ServerIron selects the server that has the fewest active session on the WSM CPU attached to the real server. The number of sessions is updated when session entries are deleted.</li> </ul>	Least connections	6-55
Symmetric SLB priority	In Symmetric SLB configurations, indicates whether this ServerIron is the default active ServerIron for the virtual server or is a standby.	Zero (disabled)	6-55

**Table 6.1: SLB Parameters (Continued)**

Parameter	Description	Default	See page...
Track ports	Groups this port as a “primary” port with up to four additional application ports. After the ServerIron sends a client request for the primary port to a real server, subsequent requests from the client for ports grouped with the primary port go to the same real server.	None configured	6-56
Track port group	Groups up to eight application ports together. After the ServerIron sends a client request for any of the grouped ports to a real server, subsequent requests from the client for ports in the group go to the same real server.	None configured	6-57
Server cluster support	Configures the ServerIron to stop sending traffic on an established connection to a server when the requested application is down on the server. This feature is useful in server cluster configurations such as a Network File System (NFS) server farm.	Disabled	6-59
Fast aging for UDP sessions	Immediately removes a UDP session entry once a reply to a client’s request is received.  <b>Note:</b> Session entries for the well-known DNS and RADIUS ports are immediately removed when a server reply is received.	Disabled	6-59
Normal aging for DNS or RADIUS	Uses the UDP age timer to delete session entries for the well-known DNS or RADIUS ports.  Other UDP ports already use the UDP age timer.	Sessions are immediately removed when server reply is received	6-60
Transparent VIP	The ServerIron load balances requests for this virtual server but does not “own” the virtual server. Multiple ServerIrons on which this virtual server is configured to be transparent can load balance requests for the server.	Disabled	6-60
<b>Virtual Server Application Port Parameters</b>			
Port state	Whether the port is enabled or disabled.	Enabled	6-60
Sticky	Whether the ServerIron sends all requests from the same client to this application to the same real server during the current session.	Disabled	6-61
Concurrent	Whether the port is configured for concurrent connections. A port configured to allow concurrent connections can have more than connection open to the same client at the same time.	Disabled	6-61
SwitchBack (DSR)	Enables the port for SwitchBack. In this type of configuration, the ServerIron does not expect server replies for the application to pass back through the ServerIron. The ServerIron changes the way it sends health checks to the application so that the health checks do not rely on the return traffic.	Disabled	6-62

Table 6.1: SLB Parameters (Continued)

Parameter	Description	Default	See page...
Smooth factor	A value used by the server response time and weighted load balancing methods to smooth out multiple response time samples for a real server.	90	6-62
Stateless	Whether the ServerIron creates a session table entry for sessions with the application. The session table entries allow the ServerIron to maintain state information for the application.	Disabled  The ServerIron does create session table entries by default.	6-64
Virtual Source	Configures session persistence in a proxy environment, where client requests arrive at the ServerIron from a proxy that assigns an IP address to each client request as it arrives at the proxy. The ServerIron sends all client traffic from a specified range of IP addresses to the same real server for the application ports you specify.	Disabled	6-64
Translation	Whether the ServerIron translates the application port number on the real server into the application port number on the virtual server in a port binding.	Enabled	6-65
		Disabled	6-65

## Configuration Guidelines

The following configuration guidelines should be observed when configuring SLB on a switch:

- Each virtual server name and IP address must be unique.
- Each virtual server can have multiple TCP/UDP ports assigned.
- Each real server name and IP address must be unique.
- Each real server can have multiple TCP/UDP ports assigned.
- Each real server TCP/UDP port can be bound only to one virtual TCP/UDP port. One virtual TCP/UDP port can be bound to one or more real TCP/UDP ports.

**NOTE:** If you need to map a real server port to multiple VIP ports, you can use the many-to-one TCP/UDP port binding feature. See “Many-To-One TCP/UDP Port Binding” on page 6-98.

- The selection of a real server among many is managed by the selected predictor (load balancing metric).
- Binding must be done to establish a relationship between virtual and real servers.

## Basic Configuration Example

A basic SLB configuration consists of a single ServerIron and multiple real servers with identical content.

To configure basic SLB:

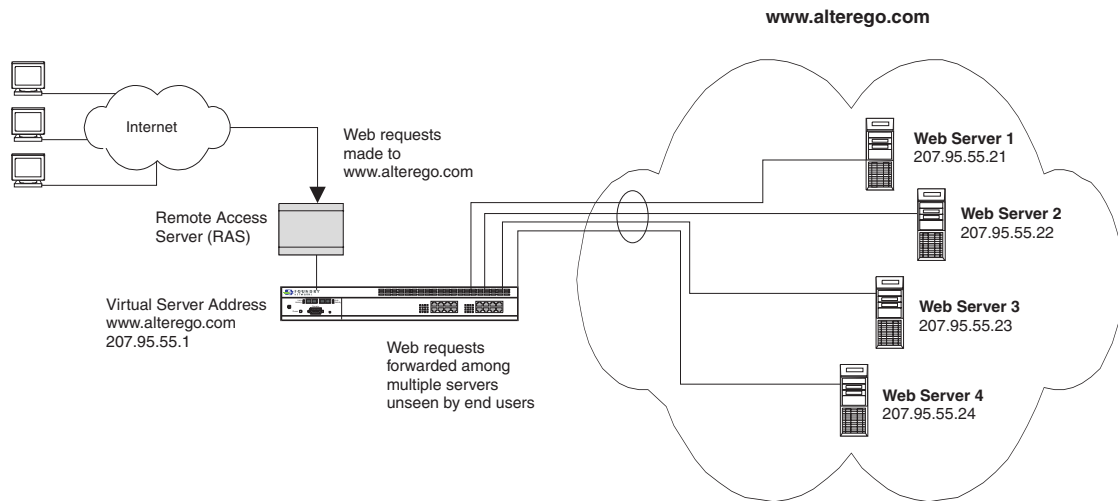
- Define the real servers on the ServerIron. For each real server, identify the TCP or UDP application ports for which you want the ServerIron to balance client traffic. The real servers contain the content you are load balancing.
- Define a virtual server (VIP). The VIP is the IP address or server name to which client browsers send requests. Add the TCP or UDP application ports the ServerIron will load balance. These should be the same application ports you specified for the real servers.

- Bind the real servers to the VIP. The bindings are based on the TCP and UDP application ports you are load balancing.

Figure 6.1 shows an example of a basic SLB configuration. This example uses multiple web servers to handle remote web requests received on the web site. The web site URL is assigned to the switch as the focal point for all inquiries as a virtual server address. The virtual server will then forward requests to each of the four web servers as specified by the predictor (load balancing metric).

The sections following the example show how to configure the ServerIron in the example using the CLI and the Web management interface.

**Figure 6.1 Basic SLB configuration**



**Table 6.2: Real and virtual server assignments**

Domain Name	Virtual IP	Port	Real IP	Port
www.alterego.com	207.95.55.1	80	207.95.55.21 (web1)	80
			207.95.55.22 (web2)	80
			207.95.55.23 (web3)	80
			207.95.55.24 (web4)	80

## Defining the Real Servers and Adding the Application Ports

To define the real servers and the TCP/UDP ports shown in Figure 6.1, use one of the following methods.

### USING THE CLI

```
ServerIron(config)# server real web1 207.95.55.21
ServerIron(config-rs-web1)# port http
ServerIron(config-rs-web1)# server real web2 207.95.55.22
ServerIron(config-rs-web2)# port http
ServerIron(config-rs-web2)# server real web3 207.95.55.23
ServerIron(config-rs-web3)# port http
ServerIron(config-rs-web3)# server real web4 207.95.55.24
ServerIron(config-rs-web4)# port http
```

**Syntax:** [no] server real-name <text> <ip-addr>



**Syntax:** [no] port <tcp/udp-port>

These commands have additional, optional parameters, described in the following sections:

- “Configuring Real Server Parameters” on page 6-37
- “Configuring Real Server Application Port Parameters” on page 6-49

---

**NOTE:** If a real server is not reachable from the ServerIron at Layer 2 (does not respond to ARP requests), and if the router connecting the ServerIron to the real server is not running proxy ARP, use the following command instead:

**server remote-name** <name> <ip-addr>

This command adds the server as a remote server. See “Web Hosting with Geographically-Distributed Servers” on page 6-110 for information.

---

**NOTE:** If the ServerIron and real server are in different sub-nets, you might need to enable source NAT and configure a source IP address. See “Web Hosting with ServerIron and Real Servers in Different Sub-Nets” on page 6-107.

---

**NOTE:** If you plan to configure SLB for a range of contiguous IP addresses on the server starting with the IP address you entered above, enter the following command at the Real Server configuration level:

**host-range** <range>

See “Web Hosting with Unlimited Virtual IP Addresses” on page 6-101 for information.

---

#### *USING THE WEB MANAGEMENT INTERFACE*

To define the real servers and the TCP/UDP ports shown in Figure 6.1:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu. The panel shown in Figure 6.2 will appear

Figure 6.2 General SLB parameters

General	
TCP Sync Limit:	65535
Ping Retries:	4
TCP Age:	30
Reassign Threshold:	20
TCP syn-def:	0
Backup preference:	5
ICMP message:	<input type="checkbox"/>
Source NAT:	<input type="checkbox"/>
Sticky Age:	5
Max ssl session id:	8192
Load Balancing Metric:	<input checked="" type="radio"/> Least Connection <input type="radio"/> Round Robin <input type="radio"/> Weighted
Max Session Limit:	524288
Ping Interval:	2
UDP Age:	5
Force Shutdown:	<input type="checkbox"/>
Clock Scale:	0
Backup timer:	10
L4 check:	<input checked="" type="checkbox"/>
Reverse NAT:	<input type="checkbox"/>
Session-id Age:	30
Max URL switch:	100000

**NOTE:** The opening panel for SLB is the configuration point for all global parameters. From this panel, you can access other panels to configure virtual servers, real servers, and server ports, and you can bind the ports on the virtual and real servers together.

5. Select the Real Server link at the bottom of the Server Load Balancing configuration sheet. The panel shown in Figure 6.3 will appear.

Figure 6.3 Real Server entry panel

The screenshot shows the Foundry Networks Device Management web interface in a Microsoft Internet Explorer browser window. The address bar shows the URL <http://209.157.20.2/>. The left sidebar contains a tree view with the following items: Port, VLAN, STP, Trunk, Static Station, SLB, General, Backup, Bind, Real Server (selected), Real Server Port, Router Interface, Source IP, TCP/UDP Port, and Virtual Server. The main content area is titled "Real Server" and contains the following configuration fields:

Server Name:	
Server IP:	0.0.0.0
Maximum Connections:	1000000
Weight:	1
Host Range:	1
Remote:	<input type="checkbox"/>
Source NAT:	<input type="checkbox"/>

Below the configuration fields are buttons for , , , and . A  button is also present. At the bottom of the panel, there are links for [\[Virtual Server\]](#), [\[Virtual Server Port\]](#), [\[Real Server Port\]](#), and [\[Bind\]](#). At the very bottom of the interface, there are links for [\[Home\]](#), [\[Site Map\]](#), [\[Logout\]](#), [\[Save\]](#), [\[Frame Enable\]](#), [\[Disable\]](#), and [\[TELNET\]](#).

6. Enter the name of the real server. In this case, enter one of the following names: web1, web2, web3, or web4.
7. Enter the IP address of the real server.
8. Modify the maximum connections variable if a limit of less than one million is desired. Possible values are 1 – 1,000,000 with a default of 1,000,000.
9. Enter a weight for the server, if you want to use the weighted load balancing metric and if you want to use a value greater than the default value of 1. Possible values are 1 – 64000.
10. If you plan to configure SLB for a range of contiguous IP addresses on the server starting with the IP address you entered above, enter the number of addresses in the Host Range field. See “Web Hosting with Unlimited Virtual IP Addresses” on page 6-101 for information.
11. If the server is not reachable from the ServerIron at Layer 2 (does not respond to ARP requests), and if the router connecting the ServerIron to the real server is not running proxy ARP, select the Remote checkbox to configure the server as a remote server. See “Web Hosting with Geographically-Distributed Servers” on page 6-110 for information.
12. If the ServerIron and real server are in different sub-nets, you might need to enable source NAT by selecting the Source NAT checkbox. See “Web Hosting with ServerIron and Real Servers in Different Sub-Nets” on page 6-107.
13. Select the Add button to assign the changes.
14. Repeat steps 6 – 13 for each of the real servers to be configured.
15. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device’s flash memory.

To add the real server port(s):

1. Select the Real Server Port link from the bottom of the General SLB panel or another SLB panel.
2. Select the Add Real Server Port link to display the Real Server Port configuration panel, as shown in Figure 6.4.

Figure 6.4 Real server port entry panel

The screenshot shows the 'Real Server Port' configuration panel in the Foundry Networks Device Management web interface. The left sidebar displays a tree view with 'ServerIron' expanded, and 'Real Server Port' selected under 'Port'. The main panel contains the following fields and sections:

- Server Name:** test
- TCP/UDP Port:** HTTP (with a 'User Define' button)
- Status:** ☐ Disable ☒ Enable
- Keep Alive:** ☐
- DNS Parameters:**
  - +DNS Zone:
  - +Addr Query:
  - +Proxy: ☐
- HTTP Parameters:**
  - \*Method: HEAD
  - \*URL:
  - \*Status Code:
- Group Id Range:**

From	To

Buttons at the bottom: Add, Modify, Delete, Reset.

**NOTE:** Some of the fields on the Real Server Port panel apply only to HTTP, while other fields apply only to DNS.

3. Select the TCP/UDP port type. In this example, real servers web1, web2, web3, and web4 are supporting HTTP traffic, so select HTTP.

**NOTE:** You also can define your own port rather than selecting one from the Port pulldown menu by selecting the User Define button.

4. Configure additional port parameters if needed. Most of the optional parameters are for keepalive health checking. See "Configuring Health Checks" on page 12-1. The Group ID Range fields are for URL switching. See "Configuring URL Switching" on page 11-1.
5. Select the Add button to assign the port to the server.
6. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.
7. Repeat steps 3 – 6 for each of the application ports you want to assign to the real server.
8. Repeat steps 1 – 7 for each real server.

### Cloning a Real Server

To simplify configuration for large server farms, you can clone real servers. When you clone a real server, you make a copy of the real server's configuration information under a new name. The copy includes the port bindings to the virtual server.

To clone a real server, enter commands such as the following:

```
ServerIron(config)# server real rs1 1.2.3.4
ServerIron(config-rs-rs1)# clone-server rs2 5.6.7.8
```

The first command changes the CLI to the configuration level for the real server you want to copy. The second command creates a clone of real server rs1. The clone is named "rs2" and has IP address 5.6.7.8.

**Syntax:** clone-server <name> <ip-addr>

The <name> parameter specifies the name of the clone.

The <ip-addr> parameter specifies the IP address of the clone.

---

**NOTE:** To delete a server clone, you must manually edit the startup-config file to remove the command. The "no" option is not supported for this command.

---

## Defining the Virtual Server

After you define the actual web server's physical addresses (real server), you then need to configure the external web server address on the switch. The external web server is the virtual server.

### USING THE CLI

To define the virtual name and address that is the access point for the company's web site and the supported service, enter the following commands:

```
ServerIron(config-rs-web4)# server virtual www.altergo.com 207.95.55.1
ServerIron(config-vs-www.altergo.com)# port http
```

**Syntax:** [no] server virtual-name <text> [<ip-addr>]

**Syntax:** [no] port <tcp/udp-port>

These commands have additional, optional parameters, described in the following sections:

- "Configuring Virtual Server Parameters" on page 6-52
- "Configuring Virtual Server Application Port Parameters" on page 6-60

### USING THE WEB MANAGEMENT INTERFACE

To define the virtual server name and address, enter the following:

1. Select the [Virtual Server](#) link from the SLB, Real Server, or Virtual Server entry panel. The panel shown in Figure 6.5 will appear.

**Figure 6.5 Virtual server entry panel**

**NOTE:** If a virtual server is already defined on the ServerIron, the Virtual Server panel is displayed instead. In this case, select the [Add Virtual Server](#) link.

2. Enter the name of the virtual server, in this example `www.alterego.com`.
3. Enter the IP address of the virtual server. This is the address that external users will see.
4. If you are configuring a range of VIPs based on the IP address you entered above, enter the number of hosts in the Host Range field. The ServerIron automatically creates the same number of virtual IP addresses (VIPs). The host addresses are sequentially numbered beginning with the IP address you entered above. See "Web Hosting with Unlimited Virtual IP Addresses" on page 6-101 for an example of how to use a host range.
5. If you are configuring this ServerIron with other ServerIrons for Symmetric SLB, enter the priority for this VIP on this ServerIron in the Symmetric Priority field. The priority determines which ServerIron is the default active ServerIron for the VIP. Other ServerIrons are standbys although they can actively service other VIPs at the same time. If you configured a host range, the same priority applies to all the VIPs in the range. See "Configuring Symmetric SLB and SwitchBack" on page 7-1 for more information about Symmetric SLB.
6. Select the load balancing metric. To assign a metric other than the global default for a specific server, select one of the other options not already defined as the global default. You can select one of the following. See "Load Balancing Method (Predictor)" on page 6-55 for more information.
  - Default – Uses the global default for the switch. To set the global default, use the Server Load Balancing configuration sheet.
  - Least Connection – Sends new traffic to the real server that has the least connections.
  - Round Robin – Rotates through the list of real servers, sending new traffic to the next server on the list.
  - Weighted – Uses the weights you assign to the real servers to select a real server and send the traffic to it.

**NOTE:** The global predictor (load balancing metric) default applies to all servers. You can modify this on a server-by-server basis. If you want to operate with the global metric defined on the Server Load Balancing configuration sheet, then select the Default button. When a system is first started, the global default predictor is least connections.

**NOTE:** You cannot select the server response time method using the Web management interface.

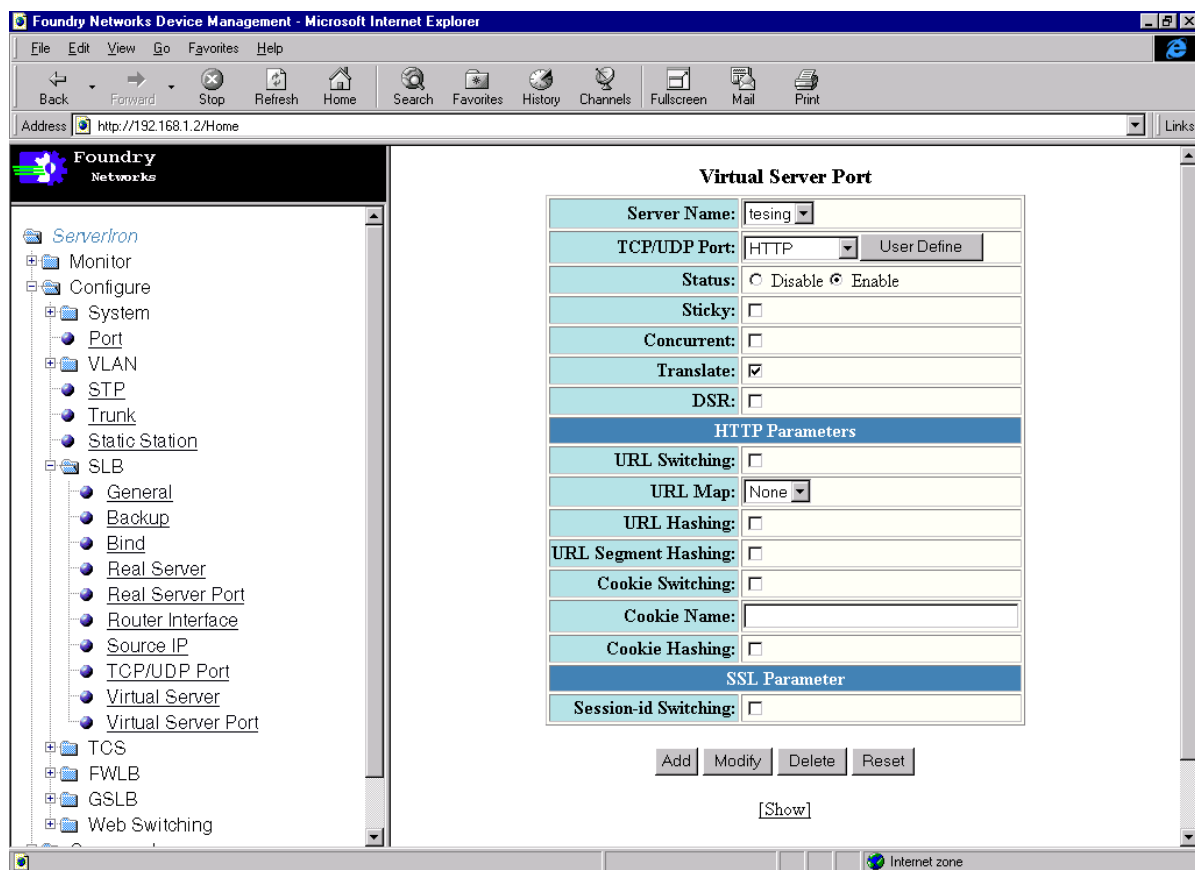
7. Select the Add button to assign the changes.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Assigning the TCP/UDP Ports

To assign the TCP/UDP ports to be supported by the virtual server:

1. Select the Virtual Server Port link from the Virtual Server entry panel.
2. Click the Modify button next to the virtual server to be modified. A panel such as the one shown in Figure 6.6 is displayed. As this example shows, all the TCP/UDP ports are enabled on the virtual server by default. To modify, disable, or re-enable individual TCP/UDP ports, use the rest of this procedure.

**Figure 6.6** Virtual Server Port panel



3. Select the TCP or UDP port (service) you want the virtual port to support from the TCP/UDP Port pulldown menu.

---

**NOTE:** You also can define your own port rather than selecting one from the Port pulldown menu by selecting the User Define button.

---

4. Enable the port if it is disabled. (If you configured a port profile for the port, then the port is already enabled.)
5. Enable the sticky option if users on this virtual port will need to reconnect to the same server.
6. Enable the concurrent option to make all secondary connections attach to the server to which the first connection attaches.
7. Disable translation if you do not want the ServerIron to translate the source IP address of responses from the real server to the client. If you disable translation, the source address in the responses will be the real server's IP address.
8. Select DSR if you want the ServerIron to use SwitchBack for the port. SwitchBack causes the real server to send responses directly to the clients, instead of sending the responses back through the ServerIron. See "Using SwitchBack" on page 7-15 for more information.
9. Select the Add button to assign the TCP/UDP port to the virtual server. If you are changing parameters for a port that you already assigned to the virtual server, select Modify.
10. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Binding Virtual and Real Servers

After you define the real servers, virtual servers, and TCP/UDP ports, you need to bind the real and virtual servers together.

To bind the four web servers shown in Figure 6.1 to the virtual server address, enter the following commands:

### USING THE CLI

```
ServerIron(config-rs-web4)# server virtual www.altergo.com
ServerIron(config-vs-www.altergo.com)# bind http web1 http
ServerIron(config-vs-www.altergo.com)# bind http web2 http
ServerIron(config-vs-www.altergo.com)# bind http web3 http
ServerIron(config-vs-www.altergo.com)# bind http web4 http
```

**Syntax:** bind <tcp/udp-port-number> <real-server-name> <tcp/udp-port-number>

---

**NOTE:** For clarity, the bindings in the example above are shown as four separate entries. Alternatively, you can enter all the binding information as one command: **bind http web1 http web2 http web3 http web4 http**

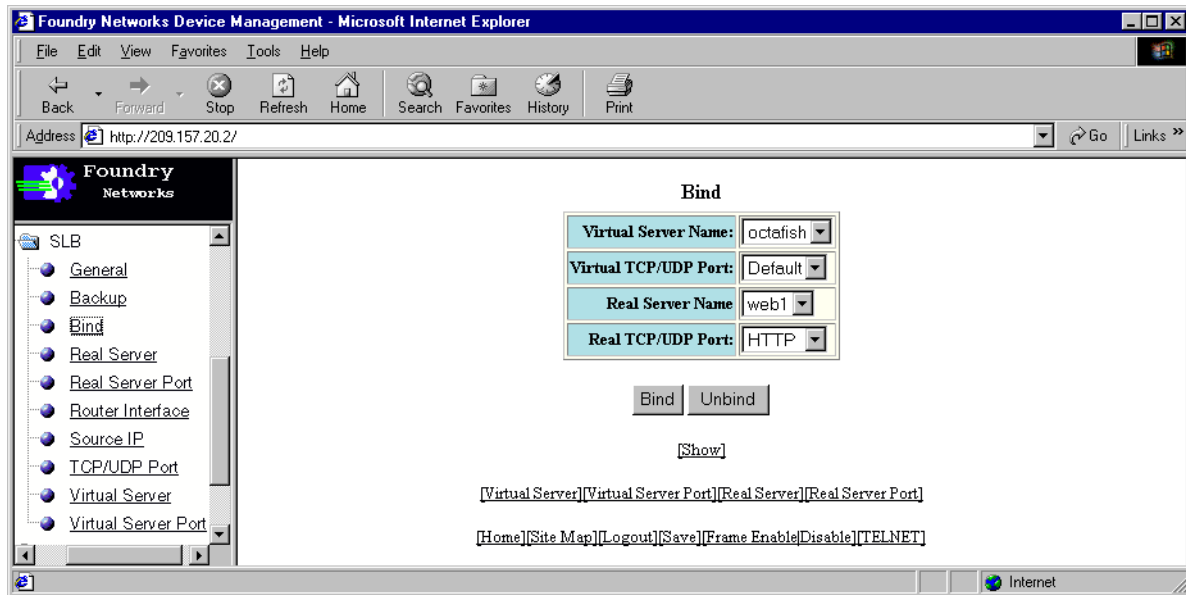
---

### USING THE WEB MANAGEMENT INTERFACE

1. Select the Bind link from the SLB, Real Server, or Virtual Server configuration panel to display the panel shown in Figure 6.7.



Figure 6.7 Entry panel to bind virtual and real server ports



2. Select the desired virtual server name from the Virtual Server Name pulldown menu.
3. Select the desired virtual server TCP/UDP port from the Virtual TCP/UDP Port pulldown menu. If you want to bind all the ports on the server, select Default.
4. Select the desired real server name from the Real Server Name pulldown menu.
5. Select the desired real server TCP/UDP port from the Real TCP/UDP Port pulldown menu. If you want to bind all the ports on the server, select Default.
6. Select the Bind button to map the selected virtual server and service to the selected real server and TCP/UDP server port.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Configuring Global SLB Parameters

The following sections describe how to configure global SLB parameters.

These parameters come with the default settings. You do not need to modify these parameters unless network needs require it. The following sections describe the parameters, their possible values, and how to configure them using the CLI or Web management interface.

---

**NOTE:** To change the maximum number of sessions, TCP age, UDP age, or reassign threshold, see "Configuring Port and Health Check Parameters" on page 12-1.

---

If you are using the Web management interface, you can access global SLB parameters from the General panel, shown in Figure 6.8. The panel also contains some global ServerIron parameters. For information, see "Configuring Port and Health Check Parameters" on page 12-1.

**Figure 6.8 SLB General configuration panel**

General			
TCP Sync Limit:	65535	Max Session Limit:	524288
Ping Retries:	4	Ping Interval:	2
TCP Age:	30	UDP Age:	5
Reassign Threshold:	20	Force Shutdown:	<input type="checkbox"/>
TCP syn-def:	0	Clock Scale:	0
Backup preference:	5	Backup timer:	10
ICMP message:	<input type="checkbox"/>	L4 check:	<input checked="" type="checkbox"/>
Source NAT:	<input type="checkbox"/>	Reverse NAT:	<input type="checkbox"/>
Sticky Age:	5	Session-id Age:	30
Max ssl session id:	8192	Max URL switch:	100000
Load Balancing Metric:	<input checked="" type="radio"/> Least Connection <input type="radio"/> Round Robin <input type="radio"/> Weighted		

## Fast-Path SLB Processing

You can enable the ServerIron to use fast-path processing for stateful or stateless SLB. When you enable fast-path processing, the ServerIron does not process every TCP or UDP packet in a given session in detail. Instead, the ServerIron uses information gathered during setup of the session to forward packets in the session.

You can enable fast-path processing for either stateful SLB or stateless SLB:

- Stateful SLB – The standard form of SLB that uses session table entries to track session information. All traffic for stateful SLB takes an optimized processing path.
- Stateless SLB – A form of SLB that does not use session table entries. All packets that go through stateless ports take an optimized processing path.

**NOTE:** SLB optimization is useful if simple SLB (stateful or stateless) is the primary or sole application on the device. If you use the ServerIron for other features such as GSLB or FWLB, SLB optimization is not useful.

**NOTE:** SLB optimization is supported only on the ServerIron 400 and ServerIron 800.

## Configuration Considerations

- You can use only one type of optimization at a time. You cannot use stateful and stateless optimization at the same time.
- Optimization applies only to SLB TCP or UDP traffic that is initiated by clients. Other types of traffic are not optimized.
- Optimization does not apply to fragmented IP packets.
- In the current release, the port name or number on the VIP must be same as the one on the real server bound to the VIP. Port translation is not supported.
- FTP traffic is not supported.
- Source NAT (**source-nat** command) is not supported.
- Host ranges (**host-range** command) are not supported.
- Many-to-one TCP/UDP port binding (**no port <tcp/udp-port> translate**) is not supported.

**NOTE:** Traffic for an SLB configuration that does not meet these criteria is still forwarded using normal processing, but fast-path processing is not used.

---

- For stateless SLB, optimization is supported only for the following TCP or UDP ports that are well-known to the ServerIron:
  - 7 (echo)
  - 9 (discard)
  - 22 (ssh)
  - 23 (telnet)
  - 25 (smtp)
  - 37 (time)
  - 49 (tacacs)
  - 53 (dns)
  - 67 (bootps)
  - 68 (bootpc)
  - 69 (tftp)
  - 80 (http)
  - 109 (pop2)
  - 110 (110)
  - 119 (nntp)
  - 123 (ntp)
  - 137 (netbios-ns)
  - 138 (netbios-dgm)
  - 143 (imap4)
  - 161 (snmp)
  - 162 (snmp-trap)
  - 179 (bgp)
  - 195 (dnsix)
  - 389 (ldap)
  - 434 (mobile-ip)
  - 517 (talk)
  - 520 (rip)
  - 554 (rtsp)
  - 1558 (xing)
  - 1755 (mms)
  - 1812 (radius)
  - 1645 (radius-old)
  - 7070 (pnm)

## Enabling SLB Optimization

To enable stateful SLB optimization, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# server fast-stateful
```

**Syntax:** [no] server fast-stateful

To enable stateless SLB optimization, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# server fast-stateless
```

**Syntax:** [no] server fast-stateless

## Load Balancing Method (Predictor)

You can fine-tune how traffic is distributed across multiple real servers by selecting one of the following load balancing metrics:

- Least connections – The ServerIron sends the request to the real server that currently has the fewest active connections with clients.
- Least sessions – The ServerIron sends the request to the real server that currently has the fewest session table entries.
- Round-robin – The ServerIron sends the request to each server in rotation, regardless of how many connections or sessions each server has.
- Weighted – The ServerIron uses the weights you assign to the real servers to select a real server. The weights are based on the number of session table entries the ServerIron has for each server.
- Response time only – The ServerIron selects the real server with the fastest response time.
- Least connection and server response time weights – The ServerIron compares a combination of a real server's least-connections weight and server response time weight to the same values for the other real servers.
- Least local connections (ServerIron 400 or ServerIron 800 only) – On an individual WSM CPU basis, the ServerIron selects the real server with the fewest active connections with clients. The predictor selects the real server that has the least number of connections created by the local WSM CPU. The local WSM CPU is the CPU that is managing the chassis slot connected to the real server.
- Least local sessions (ServerIron 400 or ServerIron 800 only) – On an individual WSM CPU basis, the ServerIron selects the server that has the fewest active session on the WSM CPU attached to the real server. The number of sessions is updated when session entries are deleted.

---

**NOTE:** Foundry recommends that you enable health checking when using either of the response-time metrics. When health checking is enabled, a server's response time consists of the combination of its response to client requests and its response to Layer 4 or Layer 7 health checks from the ServerIron.

---

You can assign these metrics on a global basis and an individual virtual server basis. By default, least connections is applied globally to all virtual servers. If you define a metric for a specific virtual server, that metric takes precedence over the globally defined metric.

---

**NOTE:** The least sessions and least local sessions predictors are supported in software releases 07.1.19, 07.2.14, and 07.3.03 and later. The least local connections predictor is supported in software release 07.2.25 and later 07.2.x releases.

---

## Least Connections

The least connections method directs a service request to the server with the fewest active connections.

For sites where a number of servers have similar performance, the least connections option smooths distribution if a server gets bogged down.

For sites where the capacity of various servers varies greatly, the least connections option maintains an equal number of connections among all servers. This results in those servers capable of processing and terminating connections faster receiving more connections than slower servers over time.

### Least Sessions

The least sessions method directs a service request to the server with the fewest session table entries in the ServerIron's session table.

### Round Robin

The round robin method directs the service request to the next server, and treats all servers equally regardless of the number of connections or response time. For example, in a configuration of four servers, the first request is sent to server1, the second request is sent to server2, the third is sent to server3, and so on. After all servers in the list have received one request, assignment begins with server1 again. If a server fails, SLB avoids sending connections to that server and selects the next server instead.

### Weighted Percentage

The weighted percentage method allows you to assign a performance weight to each server. Weighted load balancing is similar to least connections, except servers with a higher weight value receive a larger percentage of connections at a time. You can assign a weight to each real server, and that weight determines the percentage of the current connections that are given to each server. The default weight is 0.

For example, in a configuration with five servers of various weights, the percentage of connections is calculated as follows:

Weight server1 = 7

Weight server2 = 8

Weight server3 = 2

Weight server4 = 2

Weight server5 = 5

Total weight of all servers = 24

The result is that server1 gets 7/24 of the current number of connections, server2 gets 8/24, server3 gets 2/24, and so on. If a new server, server6, is added with a weight of 10, the new server gets 10/34.

If you set the weight so that your fastest server gets 50 percent of the connections, it will get 50 percent of the connections at a given time. Because the server is faster than others, it can complete more than 50 percent of the total connections overall because it services the connections at a higher rate. Thus, the weight is not a fixed ratio but adjusts to server capacity over time.

### Server Response Time

The Server Response Time method selects the real server with the fastest response time. If Layer 4 or Layer 7 health checks are disabled, the response time is based on how quickly the server responds to client requests forwarded by the ServerIron. If the health checks are enabled, the response time is the combination of the response to forwarded client queries and the response to the health checks. The ServerIron calculates the response time based on TCP SYN and TCP SYN ACK packets.

---

**NOTE:** For SwitchBack (Direct Server Return) configurations, since the ServerIron does not see the server reply traffic, the ServerIron uses only the health check responses to measure the response time.

---

### Least Connection and Server Response Time Weights

The server response time method, when used by itself, always selects the real server with the fastest response time. If all your real servers have similar response capacities, then using the server response time metric by itself generally provides an even load-balancing distribution among the real servers. However, if your server farm contains a mixture of servers, some of which have greater response capability than others, you might want to set the Server Response Time weights on individual real servers.

The default server response time weight is 0 (no weight). You can specify a weight from 0 – 65000. Setting a real server's weight higher relative to other real servers biases the ServerIron's load-balancing selections toward that real server.

### Least Local Connections

On an individual WSM CPU basis, the ServerIron selects the real server with the fewest active connections with clients. The predictor selects the real server that has the least number of connections created by the local WSM CPU. The local WSM CPU is the CPU that is managing the chassis slot connected to the real server. This method applies only to the ServerIron 400 and ServerIron 800.

### Least Local Sessions

On an individual WSM CPU basis, the ServerIron selects the server that has the fewest active session on the WSM CPU attached to the real server. The number of sessions is updated when session entries are deleted. This method applies only to the ServerIron 400 and ServerIron 800.

### Changing the Load-Balancing Method (Predictor)

To globally change the load-balancing method used by the ServerIron, use either of the following methods.

---

**NOTE:** If you enable server response time load balancing, you can weight individual servers based on a combination of weight and response time. See "Setting a Real Server's Weight Based on Response Time" on page 6-49.

---

#### USING THE CLI

To change the predictor, enter a command such as the following:

```
ServerIron(config)# server predictor round-robin
```

The command in the example above changes the predictor to round robin.

**Syntax:** [no] server predictor least-conn | least-local-conn | least-local-sess | response-time | round-robin | weighted

---

**NOTE:** If you enable the weighted percentage method, you must configure both the virtual and real servers involved. Each real server is assigned a weight from 0 – 64000. The default weight is 0.

---

#### USING THE WEB MANAGEMENT INTERFACE

To modify the predictor:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu. The panel shown in Figure 6.8 will appear.
5. Select the desired predictor.
6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

---

**NOTE:** When you assign the weighted percentage metric, you must configure both the virtual and real servers involved. Each real server is assigned a weight from 1 – 64000.

---

## Router Ports

If the ServerIron is attached to multiple routers or to a single router configured for VRRP, FSRP, or HSRP, you need to identify the ports on the ServerIron that are attached to the router(s). Explicitly identifying the ports enables the ServerIron or switch to handle Layer 4 traffic correctly.

### USING THE CLI

To identify port 8 on a ServerIron as a router port, enter the following command:

```
ServerIron(config)# server router-port 8
```

**Syntax:** server router-ports <portnum>

---

**NOTE:** To define multiple router ports on a switch, enter the port numbers, separated by blanks. You can enter up to eight router ports in a single command line. To enter more than 8 ports, enter the **server router-port...** command again with the additional ports.

---

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the Router Interface link from the menu.
5. Select the boxes next to those ports that are connected to routers.
6. Select the Apply button to assign the configuration.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## TCP SYN Limit

You can limit the maximum number of TCP SYN requests on a per-second basis per server. A TCP SYN request is a packet a client sends requesting a TCP connection to the server. Possible values are 1 – 65535. The default value is 65535.

### USING THE CLI

To limit the connections to a maximum of 3500 for all web servers on the network shown in Figure 6.1, enter the following command:

```
ServerIron(config)# server syn-limit 3500
```

**Syntax:** [no] server syn-limit <1 – 65535>

### USING THE WEB MANAGEMENT INTERFACE

To modify the TCP SYN connections supported for all web servers shown in Figure 6.1:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu. The panel shown in Figure 6.8 will appear.
5. Enter a value from 1 – 65535 in the TCP Sync Limit field.
6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Warning and Shutdown Thresholds

Response-time thresholds for real servers enable you to display warning messages when a server's response is too slow and even to stop using the server. You can specify a warning threshold and a shutdown threshold:

- **Warning** – If an application's average response time is longer than the number of milliseconds of the warning threshold, the software generates a Syslog message and an SNMP trap.
- **Shutdown** – If an application's average response time is longer than the number of milliseconds of the shutdown threshold, the software generates a Syslog message and an SNMP trap and also shuts down the application port on the real server. Other application ports on the real server are not affected.

By default, a real server does not have a warning threshold or a shutdown threshold. For each threshold, you can specify a threshold value from 0 (disabled) – 65535 milliseconds (65 seconds).

You can configure one or both thresholds globally or on an individual real server basis. The thresholds configured on an individual real server override the globally configured thresholds. After bringing down the application port, the ServerIron periodically attempts to reach the port and brings the port back up once the port responds. For information, see "Application Port States" on page 12-16.

---

**NOTE:** This feature requires the Layer 4 and Layer 7 health checks to be enabled. If the health checks are not enabled, the ServerIron does not apply the response thresholds you configure.

---

---

**NOTE:** This feature applies only to TCP ports.

---

## Globally Configuring Warning and Shutdown Thresholds

To globally configure the warning and shutdown thresholds for all real servers, enter a command such as the following at the global CONFIG level of the CLI:

```
ServerIron(config)# server response-time 200 300
```

The command in this example configures the ServerIron to generate a warning message for an application port if its average response time is longer than 200 milliseconds. The command also configures the ServerIron to shut down a port if its average response time is longer than 300 milliseconds.

**Syntax:** [no] server response-time <warning-threshold> [<shutdown-threshold>]

The <warning-threshold> parameter specifies the average number of milliseconds within which an application port must respond to avoid a warning message. You can specify from 0 – 65535 milliseconds (65 seconds). There is no default. If you specify 0, the warning threshold is disabled.

The <shutdown-threshold> parameter specifies the average number of milliseconds within which an application port must respond to avoid being shut down. You can specify from 0 – 65535 milliseconds (65 seconds). There is no default. If you specify 0, the shutdown threshold is disabled.

If you want the ServerIron to generate a warning message but you do not want the ServerIron to shut down an application port, configure the warning threshold but not the shutdown threshold. Here is an example:

```
ServerIron(config)# server response-time 100
```

To set the shutdown threshold without also setting a warning threshold, enter 0 for the warning threshold, as shown in the following example:

```
ServerIron(config)# server response-time 0 300
```

## Configuring Warning and Shutdown Thresholds for an Individual Real Server

See "Configuring Warning and Shutdown Thresholds for an Individual Real Server" on page 6-46.

## Viewing Threshold Messages in the Syslog

When an application port's average response time exceeds the warning or shutdown threshold, the ServerIron generates a Syslog message and an SNMP trap.



To display Syslog entries, enter the following command at any level of the CLI:

```
ServerIron# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 5 overruns)
  Buffer logging: level ACDMEINW, 50 messages logged
  level code: A=alert C=critical D=debugging M=emergency E=error
               I=informational N=notification W=warning
  Log servers: IP 10.10.10.20, Port 514

Dynamic Log Buffer (50 entries):
00d00h13m06s:I:running-config was changed from console
00d00h08m35s:N:L4 server 10.10.10.20 r20 port 80 is up
00d00h08m34s:N:L4 server 10.10.10.20 r20 port 80 is down
00d00h08m34s:W:Port 80 on server r20: 10.10.10.20: Avg response time 27 exceeded lower threshold
00d00h08m34s:W:Port 80 on server r20: 10.10.10.20: Avg response time 27 exceeded upper threshold; Bringing down the port...
```

The first message shown in bold type is a warning message. The last message shown in bold type is a shutdown message.

**Syntax:** show logging

## ICMP Unreachable Messages

By default, if a client requests a TCP/UDP port that is not available, the ServerIron does not send an ICMP "Destination Unreachable" message to the client. For HTTP traffic, you can configure the ServerIron to send such a message to the client by enabling the ICMP message feature. When this feature is enabled, the ServerIron sends an ICMP "Destination Unreachable" message to the client if the requested port either is not configured on any of the real servers or is unavailable because all the servers configured with the requested port are busy or down.

The ICMP message feature enables the ServerIron to send an ICMP "Destination Unreachable" message to a client for HTTP traffic. When this feature is enabled, the ServerIron will send an ICMP "Destination Unreachable" message to a client whenever an HTTP port requested by the client is not configured on any of the real servers, or the real servers that have the requested port are busy or down.

The ICMP message feature is disabled by default.

### USING THE CLI

To enable the ICMP message feature, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# server icmp-message
```

**Syntax:** [no] server icmp-message

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu. The panel shown in Figure 6.8 will appear.
5. Select the checkbox next to ICMP message to place a checkmark in the box.
6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Sending a TCP RST or ICMP Unreachable Message to a Client

By default, if the application requested by a client is not available, the ServerIron does one of the following things:

- In software release 07.2.25 and later 07.2.x releases – Sends a TCP RST to the client.
- In other software releases (07.1.x, 07.2.24 and earlier, 07.3.x) – Quietly drops the request.

Generally, an application is not available if all the real servers that have the application are unavailable or the application is not configured on the VIP requested by the client. The ServerIron can do one of the following when a requested application is unavailable:

- Quietly drop the request
- Send an ICMP Destination Unreachable message (for UDP or TCP)
- Send a TCP RST (for TCP only) – the default action

To configure the ServerIron to send an ICMP Destination Unreachable message to a client if it requests a TCP application that is unavailable, enter the following command at the global CONFIG level of the CLI:

**Syntax:** [no] server icmp-message

To disable the ServerIron from sending a TCP RST to a client if it requests a TCP application that is unavailable, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# no server reset-message
```

**Syntax:** [no] server reset-message

---

**NOTE:** ICMP messages are enabled by default in release 07.2.25 and later 07.2.x releases. The messages are disabled by default in other releases.

---

---

**NOTE:** This command overrides the **server icmp-message** command. If the configuration contains both commands, the ServerIron sends a TCP RST instead of an ICMP message for TCP requests. For UDP requests, the device still sends ICMP messages. TCP RST does not apply to UDP.

---

## Source IP Address

In addition to the ServerIron's management IP address, you can add up to eight additional IP addresses and gateways to the ServerIron. The additional IP addresses allow you to deploy the ServerIron in multinetted environments, including the following examples:

- The ServerIron and real servers are on different sub-nets.
- The remote access server (RAS) and ServerIron are on different sub-nets.
- The border access router (BAR) and ServerIron are on different sub-nets.
- The SLB configuration uses geographically-distributed servers for failover. (See the example in "Web Hosting with Geographically-Distributed Servers" on page 6-110.)

See "Web Hosting with ServerIron and Real Servers in Different Sub-Nets" on page 6-107 for an example of the type of configuration in which you need to use this feature.

---

**NOTE:** Depending on the configuration, you might also need to enable source NAT. See "Web Hosting with ServerIron and Real Servers in Different Sub-Nets" on page 6-107. See "SLB Multinetting Using Network Address Translation (NAT)" on page 2-15 for general information about the NAT operations performed by the ServerIron.

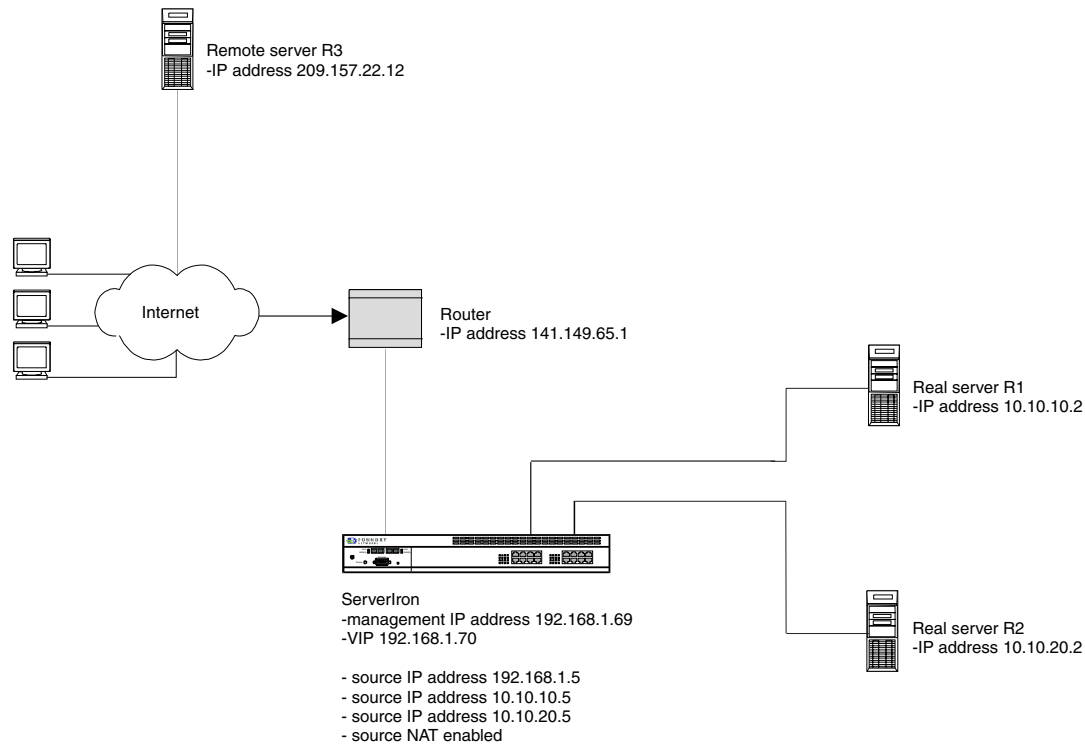
---

The ServerIron supports a maximum of 64,000 simultaneous connections on each source IP address. This maximum value is based on the architectural limits of IP itself. As a result, if you add only one source IP address, the ServerIron can support up to a maximum of 64,000 simultaneous connections to the real servers. If you configure eight source IP addresses, the ServerIron can support more simultaneous connections.

## Example

You can configure source IP addresses to enable the ServerIron to communicate with routers and real servers that are in different sub-nets than the ServerIron is in. For example, if the Figure 6.9 shows an example of a ServerIron that uses both public and private source NAT addresses.

**Figure 6.9 ServerIron configured with public and private source NAT addresses**



The ServerIron in this example is configured with three source IP addresses. Two of the addresses are in the sub-nets of the real servers and the third address is in the same sub-net as the ServerIron management address.

The software considers any address that is not within the following address ranges to be a public address. These address ranges are defined as private address ranges in RFC 1918.

- 10.0.0.0 – 10.255.255.255
- 172.16.0.0 – 172.31.255.255
- 192.168.0.0 – 192.168.255.255

## Configuring a Source IP Address

To configure a source IP address, use either of the following methods.

### USING THE CLI

To add a source IP address, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# server source-ip 192.168.1.5 255.255.255.0 209.157.22.1
```

**Syntax:** [no] server source-ip <ip-addr> <network-mask> <default-gateway>

**NOTE:** The gateway parameter is required. If you do not want to specify a gateway, enter "0.0.0.0".

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.

- Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
- Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
- Select the Source IP link from the menu. The panel shown in Figure 6.10 is displayed.

**Figure 6.10 Source IP Interface panel**

Source IP	
IP Address:	209.157.25.216
Subnet Mask:	255.255.255
Default Gateway:	0.0.0.0
Standby IP:	<input checked="" type="checkbox"/>
NAT IP:	<input type="checkbox"/>
<div>Add Modify Delete Reset</div>	

- Enter the IP address in the IP Address field.
- Enter the network mask in the Subnet Mask field.
- Optionally, enter a default gateway in the Default Gateway field, or leave "0.0.0.0" in the field.
- If this ServerIron is configured with another ServerIron in a hot standby configuration and you want the two ServerIrons to share the source IP address, select the Standby IP checkbox. When you enable this feature, the source IP address is the same regardless of which ServerIron is active.
- If you want the ServerIron to use the source IP address for source NAT, select the NAT IP checkbox.
- Click Add to implement the change.
- Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Source NAT

Source NAT allows the ServerIron to operate in a multinetted environment. You can enable source NAT globally or locally, on individual real servers. If you enable source NAT globally, the feature applies to all real servers. If you enable the feature locally, the ServerIron performs source NAT only for those real servers. Other locally-attached real servers, on which source NAT is not enabled, must be in the same sub-net as the ServerIron.

---

**NOTE:** You must also configure a source IP address. The ServerIron uses source NAT to translate its management IP address in the source field of the IP packet into the source IP address you configure. See "SLB Multinetting Using Network Address Translation (NAT)" on page 2-15 and "Source IP Address" on page 6-30.

See "Web Hosting with ServerIron and Real Servers in Different Sub-Nets" on page 6-107 for an example of the type of configuration in which you need to use Source NAT.

---

### USING THE CLI

```
ServerIron(config)# server source-nat
```

**Syntax:** [no] server source-nat

### USING THE WEB MANAGEMENT INTERFACE

To enable or disable source NAT:

- Log on to the device using a valid user name and password for read-write access.
- Click on the plus sign next to Configure in the tree view to expand the list of configuration options.

3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu. The panel shown in Figure 6.8 will appear.
5. Select Source NAT.
6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration.

## Reverse NAT

Reverse NAT allows the ServerIron to change the source IP address of some traffic initiated by a real server. Specifically, the feature causes the ServerIron to change the source IP address for traffic that the real server initiates on TCP or UDP ports that are bound to a VIP.

By default, the ServerIron does not perform address translation for any traffic initiated by the real server. However, if you enable Reverse NAT, the ServerIron does perform address translation for connections that the server initiates on ports that are bound to a VIP on the ServerIron.

Reverse NAT works with any port number you use for binding the real server to the VIP. However, TCP and UDP traffic initiated by a real server usually uses a port that is chosen by the server when the traffic is sent. As a result, it is not easy to predict the port numbers the real server will use. You can ensure that the ServerIron translates the source address of the traffic by binding the real server to a VIP using the “default” port. For example, if you configure VIP1 and bind it to real server RS1 using the default port, the ServerIron translates the source IP address in all TCP and UDP traffic initiated by RS1 from the real server’s IP address into the VIP address.

Even when Reverse NAT is enabled, the ServerIron does not translate the source address for traffic that the real server initiates over ports that are not bound to a VIP.

If you bind a real server to more than one VIP, the ServerIron will use the address of the VIP that is bound to the server using the default port. For example, if you bind a real server to VIP1 using TCP port 80 and bind the same server to VIP2 using the default port, the ServerIron always uses VIP2 for Reverse NAT.

---

**NOTE:** Reverse NAT does not affect reply traffic from the server. The feature applies only to traffic initiated by the server. In addition, the feature applies only to traffic on the TCP and UDP ports that are used to bind the real server to a VIP configured on the ServerIron. If the real server and VIP are bound using the default port, Reverse NAT applies to all TCP and UDP traffic initiated by the server.

---

Reverse NAT is disabled by default. If you need to enable reverse NAT, use one of the following methods.

### USING THE CLI

```
ServerIron(config)# server real-name R1 10.10.10.1
ServerIron(config-rs-RS1)# port http
ServerIron(config-rs-RS1)# exit
ServerIron(config)# server virtual-name VIP1 192.168.1.10
ServerIron(config-vs-VIP1)# bind http RS1 http
ServerIron(config-rs-RS1)# exit
ServerIron(config)# server virtual-name VIP2 192.168.1.69
ServerIron(config-vs-VIP1)# bind default RS1 default
ServerIron(config)# server reverse-nat
```

**Syntax:** [no] server reverse-nat

The commands in this example create real server R1 and VIPs VIP1 and VIP2. VIP1 is bound to RS1 using TCP port 80 (HTTP). VIP2 is bound to RS1 using the default port. When RS1 initiates TCP or UDP traffic, the ServerIron translates the source IP address from 10.10.10.1 to 192.168.1.69. The ServerIron uses VIP2’s IP address instead of VIP1’s IP address for Reverse NAT because VIP2 is bound using the default port.

### USING THE WEB MANAGEMENT INTERFACE

To enable or disable reverse NAT:

1. Log on to the device using a valid user name and password for read-write access.

2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu. The panel shown in Figure 6.8 will appear.
5. Select Reverse NAT.
6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Force Shutdown

SLB and TCS allow the graceful shutdown of servers and services. By default, when a service is disabled or deleted, the ServerIron does not send new connections the real servers for that service. However, the ServerIron does allow existing connections to complete normally, however long that may take.

You can use the force shutdown (sometimes called the force delete option) option to force the existing connections to be terminated within two minutes.

---

**NOTE:** If you disable or delete a service, do not enter an additional command to reverse the command you used to disable or delete the service, while the server is in graceful shutdown.

---



---

**NOTE:** See "Shutting Down a Real Server" on page 6-69 for important information about shutting down services or servers.

---

### USING THE CLI

Suppose you have unbound the Telnet service on real server 15 but you do not want to wait until the service comes down naturally. You can use the **force-delete** command to force server load-balancing connections to be terminated:

```
ServerIron(config)# server force-delete
```

**Syntax:** [no] server force-delete

### USING THE WEB MANAGEMENT INTERFACE

To enable or disable force shutdown:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu. The panel shown in Figure 6.8 will appear.
5. Select Force Shutdown.
6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Sticky Age

The sticky age is a parameter that ages out inactive sticky server connections. Possible values are from 2 – 60 minutes. The default is 5 minutes.

Sticky connections are defined on a virtual server port of an SLB switch when a service request by a client mandates a series of sequential TCP/UDP port connections to be served by the same real server. For example, if a client is accessing dynamically generated pages, the client must consistently attach to the same server, otherwise the state information will be lost.

### USING THE CLI

To modify the server sticky age to 20 from the default value of 5, enter the following command:

```
ServerIron(config)# server sticky-age 20
```

**Syntax:** [no] server sticky-age <2-60>

### USING THE WEB MANAGEMENT INTERFACE

To modify the server sticky age:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu. The panel shown in Figure 6.8 will appear.
5. Enter a value from 2 – 60 in the Sticky Age field.
6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Persistent Sticky Connections

When you unbind an application port from a server, the ServerIron temporarily places the port in the aw\_unbnd (awaiting unbind) state. If you delete an application port, the ServerIron temporarily places the port in the aw\_del (awaiting delete) state. These temporary states allow open sessions on the port to be completed before the port is unbound or removed.

By default, when the ServerIron receives a new request associated with a sticky port in the aw\_unbnd state, the ServerIron establishes the session on another real server, not the real server from which you are unbinding the port.

You can configure the ServerIron to accept new sessions for the same real server for a sticky port, even under the following conditions:

- The real server port is in the aw\_unbnd state.
- The real server port is in the aw\_del state.
- The real server port is disabled.

### USING THE CLI

To accept new connections on a real server whose sticky port has been unbound, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# server allow-sticky
```

**Syntax:** [no] server allow-sticky [refresh-age]

The **refresh-age** parameter configures the ServerIron to reset the age of a sticky session on the port whenever a new connection associated with the sticky port is established. This parameter ensures that the session stays up indefinitely until it is no longer needed.

By default, the ServerIron does not reset the age of the session when new connections are established. Instead, the session times out after the sticky age expires.

If you use the **refresh-age** parameter, the ServerIron resets the age of the session to the value of the sticky age. For example, if the sticky age is five minutes (the default), when the ServerIron establishes a new session on the sticky port, the ServerIron resets the age time for the session to five minutes. Each time the ServerIron receives another connection request associated with the sticky session, the ServerIron resets the session age again.

## Transparent VIP

Transparent VIP allows you to configure a ServerIron to transparently load balance a VIP, without owning the VIP address. This feature is useful when you want to configure multiple ServerIrons to load balance for the same VIP.

To enable transparent VIP, enable the feature globally, then configure a cache redirection policy and apply it locally to the ServerIron port(s) connected to the clients. The cache redirection policy identifies the application port(s) on the VIP that you want to load balance.

---

**NOTE:** You also must enable the feature on individual virtual servers. The feature affects only the VIPs you configure to be transparent.

---

For examples and configuration information, see “Configuring Transparent VIPs and Stateless SLB” on page 8-1.

### USING THE CLI

Enter commands such as the following to enable transparent VIP on ServerIron port 1 for TCP port 80:

```
ServerIron(config)# server transparent-vip
ServerIron(config)# ip policy 1 cache tcp 80 local
ServerIron(config)# interface ethernet 1
ServerIron(config-if-1)# ip-policy 1
```

**Syntax:** [no] server transparent-vip

**Syntax:** [no] ip policy <num> cache <tcp/udp-port> local

**Syntax:** [no] ip-policy <num>

### USING THE WEB MANAGEMENT INTERFACE

You cannot configure this feature using the Web management interface.

## TCP Fast Aging

In releases prior to 7.2.25, following a RST from the server, the ServerIron aged out session table entries in 1 – 2 minutes. Starting with release 7.2.25, following a RST from the server, the ServerIron ages out session table entries in the amount of time specified in the **server msl** command, by default 8 seconds. You can optionally configure the ServerIron to use the 1 – 2 minute aging time used in previous releases.

To set the amount of time a session table entry stays in the delete queue following a RST from the server, enter a command such as the following:

```
ServerIron(config)# server msl 2
```

**Syntax:** server msl <seconds>

The <seconds> parameter can be from 1 – 40 seconds. The default is 8 seconds.

To disable TCP fast aging, and use the 1 – 2 minute aging time from previous releases, enter the following command:

```
ServerIron(config)# server no-tcp-fast-age-on-server-reset
```

**Syntax:** no-tcp-fast-age-on-server-reset

## Decrementing the Current Connection Counter Immediately Following a Server RST

You can configure the ServerIron to immediately decrement its current connection counter when it receives a RST from the server. If a connection is maintained on two WSM CPUs, only the current connection counter on the server's WSM CPU is decremented. The current connection counter on the client's WSM CPU is not decremented immediately.

To configure the ServerIron to immediately decrement its current connection counter when it receives a RST from the server, enter the following command at the global CONFIG level of the CLI:



```
ServerIron(config)# server del-curr-conn-on-server-reset
```

**Syntax:** server del-curr-conn-on-server-reset

## Configuring Real Server Parameters

For basic real server configuration, you need to specify a name and the real server's IP address, then add the application ports that you want to load balance. The following sections describe more advanced real server options.

### IP Address

The ServerIron enables you to easily change a real server's IP address, even when the real server is active. This capability is useful when you want to perform some maintenance on the real server (either the server itself or the server's configuration on the ServerIron) or when the network topology has changed.

By default, when you change a server's IP address, the ServerIron performs the change gracefully, as follows:

- Existing connections are allowed to continue on the old IP address until they terminate normally.
- New client requests are sent to the new IP address.

Optionally, you can force all existing connections to be reset instead of waiting for them to terminate normally. When you force the connections to be reset, the ServerIron immediately resets a connection when it receives client data for the connection.

To change a real server's IP address, enter commands such as the following:

```
ServerIron(config)# server real rs1
ServerIron(config-rs-rs1)# ip-address 5.6.7.8
```

**Syntax:** [no] ip-address <ip-addr> [force-shutdown]

The <ip-addr> parameter specifies the real server's new IP address.

The **force-shutdown** parameter immediately resets a client's connection to the IP address when the ServerIron receives TCP data from the client. By default, the ServerIron allows existing connections to terminate normally following the address change.

### Location

When you define a real server, you specify whether the real server is local or remote.

- **Local** – A local server is one that is connected to the ServerIron at Layer 2. The ServerIron uses local servers for regular load balancing.
- **Remote** – A remote server is one that is connected to the ServerIron through one or more router hops. The ServerIron uses remote servers only if all the local servers are unavailable.

---

**NOTE:** If you have already defined the server but you want to change its location, you can do so using the following Web management method.

---



---

**NOTE:** To use a remote server for regular load balancing, see "Primary and Backup Servers" on page 6-53.

---

### USING THE CLI

Use one of the following commands to define the real server:

- **server real-name** <name> <ip-addr> – Configures a local server
- **server remote-name** <name> <ip-addr> – Configures a remote server

See "Defining the Real Servers and Adding the Application Ports" on page 6-12.

## USING THE WEB MANAGEMENT INTERFACE

---

**NOTE:** You cannot change the server location using the CLI unless you remove the real server definition completely, then re-add it. If you use the Web management interface, you do not need to remove the definition.

---

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the Real Server link from the menu.
5. Select the Modify button next to the real server to be modified. The real server entry panel will appear.
6. Click on the Remote checkbox to place a checkmark in the box or remove the checkmark. If the box contains a checkmark, the server is remote and is used only as a backup.
7. Select the Modify button to assign the changes.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Backup Server

---

**NOTE:** This section does not apply to software release 07.1.x.

---

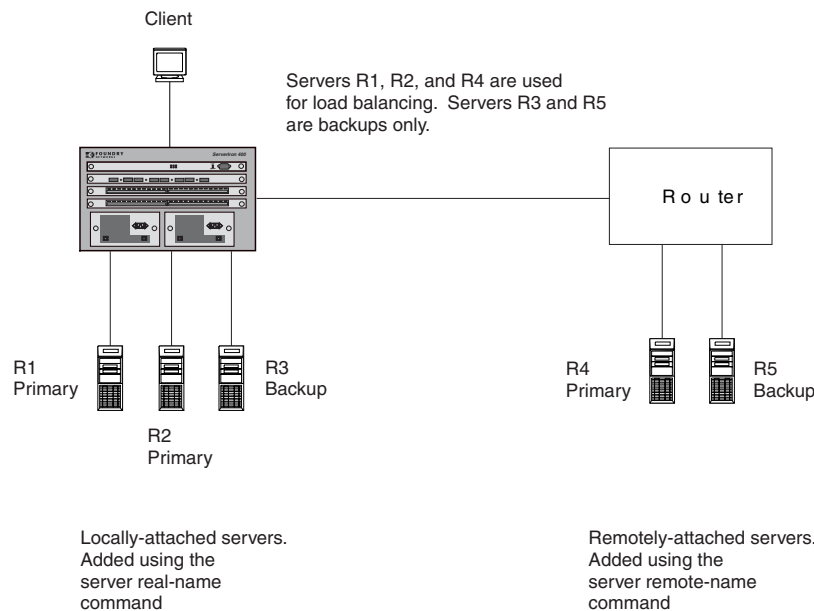
By default, the real server is either a primary server or a backup server based on how you added the server:

- Primary server (locally attached server added using the **server real-name** command or Web equivalent) – A primary server is used by the ServerIron when load balancing client requests for an application.
- Backup server (remotely attached added using the **server remote-name** command or Web equivalent) – A backup server is used by the ServerIron only if all the primary servers are unavailable for the requested application.

Using the feature described in this section, you can explicitly designate a server to be a primary or backup server, regardless of the command you used to add it. Thus, a primary or backup server can be locally attached or attached through a router.

In addition, this feature implements the primary and backup configuration on an individual VIP basis. You designate each backup server by changing the real server configurations. You do not need to designate the primary servers. You enable the feature in individual VIPs for individual application ports.

Figure 6.11 shows an SLB configuration that uses locally-attached and remotely-attached servers. The configuration also uses some of the servers as the primary load-balancing servers while using the other servers only as backups. Notice that one of the locally-attached servers is a backup server while one of the remotely-attached servers is a primary load-balancing server.

**Figure 6.11 Servers configured as primaries and backups**

By default, when this feature is enabled on a VIP and all the primary servers are unavailable, a VIP begins using the backup servers until a primary server becomes available again. Once a primary server is available, the VIP uses the primary server instead. Optionally, you can configure a VIP to continue to use the backup servers even after the primary servers become available again.

### Configuring Primary and Backup Servers

To configure primary and backup servers:

- Edit the configuration of each backup real server to designate the server as a backup.

---

**NOTE:** You do not need to designate the primary servers. The ServerIron assumes that all servers you do not designate as primary servers are backup servers.

---

- Enable use of the primary and backup servers in individual VIPs on individual application ports. Only the VIPs and application ports for which you enable the feature use it. The other application ports within the VIP, and the other VIPs, use the locally-attached servers (configured using the **server real-name** command) as their primary servers and the remotely-attached servers (configured using the **server remote-name** command) as their backup servers.

Optionally, configure the individual applications on the VIPs to continue using the backup servers following a failover, instead of returning to the primary servers.

#### Designating a Backup Server

To designate a real server to be a backup server, enter the following command at the configuration level for the server:

```
ServerIron(config-rs-R3)# backup
```

**Syntax:** [no] backup

#### Enabling a VIP to Use the Primary and Backup Servers

To enable a VIP to use the servers designated as backups only as backups, and use the other servers as the load-balancing servers, enter the following command at the configuration level for the VIP:

```
ServerIron(config-vs-VIP1)# port http lb-pri-servers
```

This command enables VIP1 to use the backup and primary servers for application port HTTP.

To configure the VIP and application port to continue using the backup servers even after the primary servers become available again, use the backup-stay-active parameter, as in the following example:

```
ServerIron(config-vs-VIP1)# port http lb-pri-servers backup-stay-active
```

**Syntax:** [no] port <tcp/udp-port> lb-pri-servers [backup-stay-active]

### Complete CLI Example

Here are the commands for implementing the load-balancing configuration shown in Figure 6.11 on page 6-39.

The following commands configure the real servers. Notice that the **backup** command is used with servers R3 and R5.

```
ServerIron(config)# server real-name R1 10.10.10.10
ServerIron(config-rs-R1)# port http
ServerIron(config-rs-R1)# exit
ServerIron(config)# server real-name R2 10.10.10.20
ServerIron(config-rs-R2)# port http
ServerIron(config-rs-R2)# exit
ServerIron(config)# server real-name R3 10.10.10.30
ServerIron(config-rs-R3)# backup
ServerIron(config-rs-R3)# port http
ServerIron(config-rs-R3)# exit
ServerIron(config)# server remote-name R4 198.10.10.40
ServerIron(config-rs-R4)# port http
ServerIron(config-rs-R4)# exit
ServerIron(config)# server remote-name R5 198.10.10.50
ServerIron(config-rs-R5)# backup
ServerIron(config-rs-R5)# port http
```

The following commands configure the VIP.

```
ServerIron(config-rs-R5)# server virtual-name VIP1 198.10.10.100
ServerIron(config-vs-VIP1)# port http lb-pri-servers
ServerIron(config-vs-VIP1)# bind http R1 http R2 http R3 http R4 http R5 http
```

### Application Ports

You must specify the application ports that you want the ServerIron to load balance. The ServerIron sends client requests only to the application ports you specify in the real server definition.

You can use the CLI or the Web management interface to add an application port to a real server you have already defined.

#### USING THE CLI

To add application ports to a real server, enter commands such as the following:

```
ServerIron(config)# server real web1 207.95.55.21
ServerIron(config-rs-web1)# port http
ServerIron(config-rs-web1)# port ftp
ServerIron(config-rs-web1)# port 8080
```

**Syntax:** [no] port <tcp/udp-port>

This command has additional, optional parameters. See “Configuring Real Server Application Port Parameters” on page 6-49.

#### USING THE WEB MANAGEMENT INTERFACE

1. Select the [Real Server Port](#) link from the bottom of the General SLB panel or another SLB panel.
2. Select the [Add Real Server Port](#) link to display the Real Server Port configuration panel.

---

**NOTE:** Some of the fields on the Real Server Port panel apply only to HTTP, while other fields apply only to DNS.

---

3. Specify the port's well-known name or number.
  - To specify a port by its well-known name, click the System Define button to activate the TCP/UDP Port field's pulldown menu, then select the name from the menu.
  - To specify a port by its number, click the User Define button to activate the input field, then enter the name from the menu.

---

**NOTE:** If the button says User Define, the menu of well-known names is activate. If the button says System Define, the port number entry field is active instead. The menu contains the names only for the ports that are known to the ServerIron.

---

4. Configure additional port parameters if needed. Most of the optional parameters are for keepalive health checking. See "Configuring Health Checks" on page 12-1. The Group ID Range fields are for URL switching. See "Configuring URL Switching" on page 11-1.
5. Select the Add or Modify button to assign the port or the changes to the port to the server.
6. Repeat steps 1 – 5 for each of the real servers to be assigned.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Host Ranges and Host-Range Maps

If you want to use the Unlimited VIP feature to load balance a large set of contiguous IP addresses on the real server, define a host range. Defining a host range simplifies configuration by allowing you to enter a single command or Web option for the whole range of addresses instead of entering information for each address individually.

For a complete configuration example, see "Web Hosting with Unlimited Virtual IP Addresses" on page 6-101.

### USING THE CLI

To configure a host range on a real server, enter commands such as the following:

```
ServerIron(config)# server real-name r1 10.0.1.6
ServerIron(config-rs-r1)# host-range 20
```

**Syntax:** [no] host-range <num>

This command configures a range of 20 IP addresses, from 10.0.0.1 through 10.0.0.20.

### USING THE WEB MANAGEMENT INTERFACE

1. Select the Real Server link from the bottom of the General SLB panel or another SLB panel.
2. Click Modify next to the row of information about the real server you want to modify. If you are adding a new real server, select the Add Real Server link instead.
3. Edit the number in the Host Range field to specify how many contiguous IP addresses you want to configure based on the real server's IP address.
4. Select the Add or Modify button to assign the port or the changes to the port to the server.
5. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Host-Range Maps

A host range allows you to easily configure a contiguous range of VIP addresses. Instead of individually configuring each VIP address, you can configure the base VIP address (the lowest VIP address in the range), then specify how many addresses the range contains. These VIP addresses can, in turn, be mapped to a range of real

server addresses. When a client requests an address in the VIP host range, the ServerIron automatically maps the VIP address to a real IP address on a real server, based on the real server address's offset from the base VIP address.

For example, you can specify that a host range of 5 VIP addresses on a virtual server be mapped to a host range of 5 IP addresses on a real server. If the virtual server's base IP address is 192.168.9.10 and the real server's base IP address is 10.10.10.30, the mapping would be as follows.

Virtual Server VIP addresses	Offset from VIP base address	Real Server IP addresses
192.168.9.11	1	10.10.10.31
192.168.9.12	2	10.10.10.32
192.168.9.13	3	10.10.10.33
192.168.9.14	4	10.10.10.34

Additionally, you can map a host range of VIP addresses to a host range of IP addresses on multiple real servers. For example:

Virtual Server VIP addresses	Offset from VIP base address	Real Server 3 IP addresses	Real Server 2 IP addresses	Real Server 1 IP addresses
192.168.9.11	1	10.10.10.71	10.10.10.51	10.10.10.31
192.168.9.12	2	10.10.10.72	10.10.10.52	10.10.10.32
192.168.9.13	3	10.10.10.73	10.10.10.53	10.10.10.33
192.168.9.14	4	10.10.10.74	10.10.10.54	10.10.10.34

With host ranges, the mapping between the host range on the virtual server and the host range on the real server(s) had to be sequential and contiguous. With the **host-range map** feature, addresses in the host range on the real server(s) do not need to be contiguous.

The host-range map feature allows you to select the addresses in a real server's host range that can be mapped to addresses in the virtual server's host range. For example, using this feature, you can establish the following mapping between a host range of VIP addresses on a virtual server and a host range of IP addresses on three real servers.

**Table 6.3: VIP-to-IP address mapping using the host-range map feature**

Virtual Server VIP addresses	Offset from VIP base address	Real Server 3 IP addresses	Real Server 2 IP addresses	Real Server 1 IP addresses
192.168.9.11	1		10.10.10.51	
192.168.9.12	2	10.10.10.72	10.10.10.52	10.10.10.32
192.168.9.13	3	10.10.10.73		10.10.10.33
192.168.9.14	4		10.10.10.54	10.10.10.34

In this example, real server 1 can use addresses in its host range that are offset by 2, 3, and 4 from its base IP address to map to VIP addresses that are offset by 2, 3, and 4 from the virtual server's base VIP address. However, the IP address in real server 1's host range that is offset by 1 from its base IP address would not be mapped to the VIP address that is offset by 1 from the virtual server's base VIP address.

You can use the host-range map feature with up to 32 real servers and host ranges of up to 255 addresses.

To use the host-range map feature to establish a mapping structure like the one shown in Table 6.3, you perform the following tasks:

1. Assign a unique bind-ID to each real server bound to the virtual server. Each real server must have its own bind-ID.
2. Define a host-range map, which associates each offset in a virtual server's host range with one or more bind-IDs.
3. Apply the host-range map to the virtual server.

These steps are described in the following sections.

#### **Assigning a Bind-ID to a Real Server**

A **bind-ID** is a number you assign to a real server. When you configure the host range map, you refer to the real servers by their bind-IDs. Assign a bind-ID to each real server to be included in a host-range map.

For example, to implement the sample configuration in Table 6.3, you can assign real server 1 to bind-ID = 1, real server 2 to bind-ID = 2, and real server 3 to bind-ID = 3. The following commands configure these three real servers.

```
ServerIron(config)# server real rs1 10.10.10.30
ServerIron(config-rs-rs1)# host-range 5
ServerIron(config-rs-rs1)# bind-id 1
ServerIron(config-rs-rs1)# port http
ServerIron(config-rs-rs1)# exit

ServerIron(config)# server real rs2 10.10.10.50
ServerIron(config-rs-rs2)# host-range 5
ServerIron(config-rs-rs2)# bind-id 2
ServerIron(config-rs-rs2)# port http
ServerIron(config-rs-rs2)# exit

ServerIron(config)# server real rs3 10.10.10.70
ServerIron(config-rs-rs3)# host-range 5
ServerIron(config-rs-rs3)# bind-id 3
ServerIron(config-rs-rs3)# port http
ServerIron(config-rs-rs3)# exit
```

**Syntax:** [no] host-range <number-of-addresses>

**Syntax:** [no] bind-id <number>

The **host-range** <number-of-addresses> command specifies the number of IP addresses that will be included in the host range for the real server. For example, since real server rs1 has a base IP address of 10.10.10.30, the **host-range 5** command causes addresses 10.10.10.30 through 10.10.10.34 to be included in the host range. You use the host range map to select individual addresses within the range and omit the addresses you want to omit.

The **bind-id** <number> command specifies the bind-ID applied to the real server. The bind-ID for each real server must be unique.

#### **Defining a Host-Range Map**

The host-range map specifies which IP addresses in the host ranges of each real server you actually want to use for SLB. The map enables you to selectively include individual addresses, by specifying their offsets in the range.

To define a host range map, you associate each VIP offset with one or more bind-IDs, then determine the binary representation of this association, then convert the binary representation to a hexadecimal number. You enter this hex number as part of the host-range map definition.

When defining a host-range map, it may be useful to create a table containing a row for each VIP offset and a column for each bind-ID (real server), as well as a column for the binary representation and a column for the hex

number. For each VIP offset, specify which bind-ID can use IP addresses in its host range to map to the VIP offset address. For the sample configuration in Table 6.3 on page 6-42, such a table would look like the following:

**Table 6.4: Determining a host-range map**

VIP Offset	Bind to Bind ID = 3	Bind to Bind ID = 2	Bind to Bind ID = 1	Binary Representation	Hex Number
1		X		010	2
2	X	X	X	111	7
3	X		X	101	5
4		X	X	011	3

The first line of the table indicates that VIP offset 1 applies only to the real server with bind-ID = 2. Only real server 2 will map the IP address in its host range that is offset by 1 to the IP address that is offset by one from the VIP's base IP address. The binary representation of this is "010", which means "not bind-ID = 3, bind-ID = 2, not bind-ID = 1". The hex representation of "010" is "2". You enter this hex number as part of the definition of the host-range map.

Using the information in Table 6.4, you would define the host-range map for the sample configuration in Table 6.3 on page 6-42 as follows:

```
ServerIron(config)# vip-host-range-map 1
ServerIron(config-vip-host-range-1)# vip-offset 1 2
ServerIron(config-vip-host-range-1)# vip-offset 2 7
ServerIron(config-vip-host-range-1)# vip-offset 3 5
ServerIron(config-vip-host-range-1)# vip-offset 4 3
ServerIron(config-vip-host-range-1)# exit
```

**Syntax:** [no] vip-host-range-map <map-number>

**Syntax:** [no] vip-offset <vip-offset-number> <hex-number>

The default behavior (without a host-range map definition) is to bind each VIP address offset from the virtual server's base address to the comparable offset address on each of the real servers. In the sample configuration, the host-range map definition for VIP offset 2 specifies that addresses from all three real servers be included in the bindings. Since this is the default behavior, the **vip-offset 2 7** command in the host-range map definition can be omitted.

### Applying the Host-Range Map to the Virtual Server

After you assign the bind-IDs to the real servers and create a host-range map, you apply the host-range map to the virtual server.

For example, to apply host-range map 1 to virtual server vs1:

```
ServerIron(config)# server virtual vs1 192.168.9.10
ServerIron(config-vs-vs1)# host-range 5
ServerIron(config-vs-vs1)# host-range-map 1
ServerIron(config-vs-vs1)# port http
ServerIron(config-vs-vs1)# bind http rs1 http rs2 http rs3 http
```

**Syntax:** [no] host-range-map <map-number>

### Using the Host-Range Map Feature in a SwitchBack Configuration

If you are using SwitchBack (sometimes called "Direct Server Return"), you configure a separate loopback interface on each real server for the VIP's base address and also for each additional address in the host range you want to use on the real server.

The ServerIron sends the client traffic to the real server without translating the destination address. The real server receives the client traffic addressed to a loopback address configured on the server and responds directly to the client.



Normally, the CLI checks for address range overlaps when you configure a real server. For example, if real server abc has management IP address 10.10.10.10 and a host range of 5, the CLI assumes that the real server also will have addresses 10.10.10.11 – 10.10.10.14. If you then try to configure real server def with management IP address 10.10.10.12, the CLI detects an address overlap, since 10.10.10.12 is within the range specified for abc, and displays an error message instead of accepting the configuration. Here is an example:

```
ServerIron(config)# server real def 10.10.10.12
duplicate IP address !!!
Error - Failed to create real server
```

The overlap check is not applicable to SwitchBack configurations since the addresses in the range are not going to be configured on the real server. For example, if the VIP is 192.168.9.10 with a range of 5, you need to configure loopback interfaces 192.168.9.10 – 192.168.9.14 on each real server. You do not need to configure a range beginning with the real server's management IP address.

For a SwitchBack configuration, if the management IP address of a real server is within the host range on another real server (even though the addresses in the range will not actually be configured on the real server), you need to disable overlap checking. To disable overlap checking, enter the following command:

```
ServerIron(config)# server no-host-range-ip-check
```

**Syntax:** [no] server no-host-range-ip-check

After you disable the range check, use the commands described in the previous section to configure the real servers, bind-IDs, VIP, and host range map.

---

**NOTE:** Do not use this command unless you are configuring a host range in a SwitchBack configuration. If the configuration is not SwitchBack, disabling overlap checking can cause the feature to work incorrectly.

---

## Maximum Connections

You can limit the maximum number of sessions the ServerIron will maintain in its session table for a real server. By setting a limit for a server, you can avoid a condition where the capacity threshold of the server is exceeded.

When a real server reaches the maximum defined connection threshold, an SNMP trap is sent. When all the real servers in a server pool reach their maximum connection threshold, additional TCP/UDP packets are dropped, and an ICMP destination unreachable message is sent.

Up to one million total sessions are supported on the ServerIron. This is also the default maximum connection value for real servers.

### USING THE CLI

To modify the maximum connections supported for a specific server, enter the following commands:

```
ServerIron(config)# server real web1
ServerIron(config-rs-web1)# max-conn 145000
ServerIron(config-rs-web4)# end
ServerIron# write mem
```

**Syntax:** [no] max-conn <1-1000000>

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the Real Server link from the menu.
5. Select the Modify button next to the real server to be modified. The real server entry panel will appear.
6. Enter a value from 1 – 1,000,000 in the maximum connections field.
7. Select the Modify button to assign the changes.

8. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Traffic Rate Threshold

You can configure a threshold for the traffic rate on a real server. When this threshold is reached, the real server is not assigned any new connections, although the real server will continue to handle previously assigned connections.

---

**NOTE:** This feature is supported only on the ServerIron 400 and ServerIron 800.

---

To set a threshold for the traffic rate on a real server:

```
ServerIron(config)# server real R 10.10.10.50
ServerIron(config-rs-R)# byte-rate-threshold 10000
```

**Syntax:** [no] byte-rate-threshold <bytes-per-second>

The ServerIron uses the number of bytes in all received and transmitted TCP and UDP packets in its calculation of the traffic rate.

## Warning and Shutdown Thresholds

Response-time thresholds for real servers enable you to display warning messages when a server's response is too slow and even to stop using the server. You can specify a warning threshold and a shutdown threshold:

- **Warning** – If an application's average response time is longer than the number of milliseconds of the warning threshold, the software generates a Syslog message and an SNMP trap.
- **Shutdown** – If an application's average response time is longer than the number of milliseconds of the shutdown threshold, the software generates a Syslog message and an SNMP trap and also shuts down the application port on the real server. Other application ports on the real server are not affected.

By default, a real server does not have a warning threshold or a shutdown threshold. For each threshold, you can specify a threshold value from 0 (disabled) – 65535 milliseconds (65 seconds).

You can configure one or both thresholds globally or on an individual real server basis. The thresholds configured on an individual real server override the globally configured thresholds. After bringing down the application port, the ServerIron periodically attempts to reach the port and brings the port back up once the port responds. For information, see "Application Port States" on page 12-16.

---

**NOTE:** This feature requires the Layer 4 and Layer 7 health checks to be enabled. If the health checks are not enabled, the ServerIron does not apply the response thresholds you configure.

---



---

**NOTE:** This feature applies only to TCP ports.

---

To globally set warning and shutdown thresholds for all real servers, see "Globally Configuring Warning and Shutdown Thresholds" on page 6-28.

### Configuring Warning and Shutdown Thresholds for an Individual Real Server

To configure warning and shutdown thresholds for an individual server, enter a command such as the following at the configuration level for the real server:

```
ServerIron(config-rs-R1)# response-time 50 75
```

This command sets the warning threshold to 50 milliseconds and the shutdown threshold to 75 milliseconds, for this real server only.

---

**NOTE:** The threshold values you configure on an individual real server override the globally configured thresholds.

---

**Syntax:** [no] response-time <warning-threshold> [<shutdown-threshold>]

The parameters are the same as the ones for the **server response-time** command.

### Viewing Threshold Messages in the Syslog

When an application port's average response time exceeds the warning or shutdown threshold, the ServerIron generates a Syslog message and an SNMP trap.

To display Syslog entries, enter the following command at any level of the CLI:

```
ServerIron# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 5 overruns)
  Buffer logging: level ACDMEINW, 50 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                 I=informational N=notification W=warning
  Log servers: IP 10.10.10.20, Port 514

Dynamic Log Buffer (50 entries):
00d00h13m06s:I:running-config was changed from console
00d00h08m35s:N:L4 server 10.10.10.20 r20 port 80 is up
00d00h08m34s:N:L4 server 10.10.10.20 r20 port 80 is down
00d00h08m34s:W:Port 80 on server r20: 10.10.10.20: Avg response time 27 exceeded lower threshold
00d00h08m34s:W:Port 80 on server r20: 10.10.10.20: Avg response time 27 exceeded upper threshold; Bringing down the port...
```

The first message shown in bold type is a warning message. The last message shown in bold type is a shutdown message.

**Syntax:** show logging

### Layer 3 Health Check

By default, when you add a real server configuration to the ServerIron, the ServerIron uses a Layer 3 health check (IP ping) to determine the server's reachability. If the real server responds to the ping, the ServerIron changes the server's state to ACTIVE and begins using the server for client requests.

You can globally disable the Layer 3 health check for local servers or remote servers. You also can disable the Layer 3 health check on individual real servers. When you disable the Layer 3 health check, the ServerIron sends an ARP request for the default gateway and makes the server's state ACTIVE as long as the ARP entry is present in the ServerIron's ARP cache.

To disable the Layer 3 health check on an individual real server, enter the following command at the configuration level for the server:

```
ServerIron(config-rs-R1)# no-l3-check
```

**Syntax:** [no] no-l3-check

This command applies to local real servers and remote real servers.

---

**NOTE:** To globally disable Layer 3 health checks for local real servers or remote real servers, see "Disabling the Layer 3 Health Check for Real Servers" on page 12-20.

---

### Source NAT

Source NAT allows the ServerIron to be in more than one sub-net. If the real server and the ServerIron are in different sub-nets and they are not connected by a router that is multinetted, enable source NAT on the real server.

If you enable source NAT on a real server, the feature applies only to the server. You also can enable source NAT globally. See "Source NAT" on page 6-32.

### USING THE CLI

```
ServerIron(config)# server real-name berto
ServerIron(config-rs-berto)# source-nat
```

**Syntax:** [no] source-nat

To enable Source NAT on a real server:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the Real Server link from the menu.
5. Select the Modify button next to the real server to be modified. The real server entry panel will appear.
6. Click on the Source NAT checkbox to place a checkmark in the box.
7. Select the Modify button to assign the changes.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Weight

Suppose you want to assign a higher weight to real server web1 to bias traffic toward that server. No other changes are made to the weights of web servers 2, 3, and 4, and they remain configured with the default weight of zero (Figure 6.1).

### USING THE CLI

```
ServerIron(config)# server virtual www.alterego.com
ServerIron(config-vs-www.alterego.com)# predictor weighted
ServerIron(config-vs-www.alterego.com)# server real web1 207.95.55.21
ServerIron(config-vs-www.alterego.com)# exit
ServerIron(config)# server real web1
ServerIron(config-rs-web1)# weight 10
```

**Syntax:** weight <least-connections-weight> [<response-time-weight>]

The <least-connections-weight> parameter specifies the real server's weight relative to other real servers in terms of the number of connections on the server. More precisely, this weight is based on the number of session table entries the ServerIron has for TCP or UDP sessions with the real server. You can specify a value from 0 – 65000. The default is 1. This parameter is required. However, if you want to use a weight value only for the Server Response Time but not for the number of connections, specify 0 for this parameter.

The <response-time-weight> parameter specifies the real server's weight relative to other real servers in terms of the server's response time to client requests sent to the server. You can specify a value from 0 – 65000. The default is 0 (disabled). This weight is applicable only when the server response time load-balancing method is enabled. See "Setting a Real Server's Weight Based on Response Time" on page 6-49.

If you enter a value for <response-time-weight>, the ServerIron adds the two weight values together when selecting a real server. If you specify equal values for each parameter, the ServerIron treats the weights equally. The number of connections on the server is just as relevant as the server's response time. However, if you set one parameter to a higher value than the other, the ServerIron places more emphasis (weight) on the parameter with the higher value. For example, if you specify a higher server response time weight than the weight for the number of connections, the ServerIron pays more attention to the server's response time than to the number of connections it currently has when considering the real server for a new connection.

### USING THE WEB MANAGEMENT INTERFACE

To modify the weight assigned to a real server:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.

3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the Real Server link from the menu.
5. Select the Modify button next to the real server to be modified. The real server entry panel will appear.
6. Enter a value from 1 – 64000 in the Weight field.
7. Select the Modify button to assign the changes.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Setting a Real Server's Weight Based on Response Time

The Server Response Time metric, when used by itself, always selects the real server with the fastest response time. If all your real servers have similar response capacities, then using the Server Response Time metric by itself generally provides an even load-balancing distribution among the real servers. However, if your server farm contains a mixture of servers, some of which have greater response capability than others, you might want to set the Server Response Time weights on individual real servers.

The default Server Response Time weight is 0 (no weight). You can specify a weight from 0 – 65000. Setting a real server's weight higher relative to other real servers biases the ServerIron's load-balancing selections toward that real server.

For example, if you bind a virtual server to three real servers, and one of the servers tends to respond less quickly than the other two but otherwise has the same connection capacity as the faster servers, you can enter commands such as the following to increase the Server Response Time weight of the faster servers:

```
ServerIron(config)# server real-name wolalak
ServerIron(config-rs-wolalak)# weight 1 5
ServerIron(config-rs-wolalak)# exit
ServerIron(config)# server real-name wuwanich
ServerIron(config-rs-wuwanich)# weight 1 5
```

This command sets the Server Response Time weight on the faster servers to 5, giving the servers more weight in terms of response time than the slower real server.

**Syntax:** [no] weight <least-connections-weight> [<response-time-weight>]

---

**NOTE:** If you use the server response time method, you also can modify the smooth factor on individual application ports. See "Smooth Factor" on page 6-62.

---

## Configuring Real Server Application Port Parameters

You can globally configure an application port by configuring a port profile for the port. When you configure a port profile, the parameters in the profile apply to all servers that include the application port. To configure a port profile, see "Configuring a Port Profile" on page 12-21.

You also can locally define some SLB port parameters on an individual real-server basis:

- State (enabled or disabled) – Ports are enabled by default.
- Keepalive health check state – Keepalive health checks are enabled if you have configured a port profile for the port and did not globally disable the health check. You can locally disable the keepalive health check for the port on a specific real server while leaving the health check globally enabled.
- Layer 7 health check parameters – For some application ports that are known to the ServerIron, you can customize the Layer 7 health checks for individual real servers.

---

**NOTE:** For the HTTP ports, you also can configure Layer 7 health checks for Transparent Cache Switching.

---

You also can configure slow-start and history parameters, which are not exclusive to SLB. See "Configuring the Slow-Start Mechanism" on page 12-62 and "Monitoring Layer 4 Statistics" on page B-3.

## Port State

Application ports are enabled by default. To disable an application port on a real server, use either of the following methods.

### USING THE CLI

To disable an individual application port:

```
ServerIron(config)# server real Sy_Borg 192.168.4.69
ServerIron(config-rs-Sy_Borg)# port http disable
```

**Syntax:** [no] port <tcp/udp-port> disable

To re-enable a port, enter commands such as the following:

```
ServerIron(config)# server real Sy_Borg 192.168.4.69
ServerIron(config-rs-Sy_Borg)# no port http disable
```

To disable all the application ports on a real server, enter the following command at the configuration level for the server:

```
ServerIron(config-rs-R1)# port disable-all
```

To re-enable all the application ports, enter the following command:

```
ServerIron(config-rs-R1)# no port disable-all
```

**Syntax:** [no] port disable-all

### USING THE WEB MANAGEMENT INTERFACE

1. Select the [Real Server Port](#) link from the bottom of the General SLB panel or another SLB panel.
2. Click the Modify button for the application port to be modified. The Real Server Port panel is displayed.
3. Click Enable or Disable next to Status.
4. Select the Modify button to save the change to the running-config.
5. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Binding State

By default, a real server's application ports remain bound to the virtual servers to which you bind them. You can unbind all of a real server's application ports from the virtual servers.

To unbind a real server's application ports, enter the following command at the configuration level for the server:

```
ServerIron(config-rs-R1)# port unbind-all
```

**Syntax:** port unbind-all

---

**NOTE:** Once you unbind the ports, you can rebind them only on an individual virtual server and port basis.

---

To re-bind an application port, you must use the **bind** command at the configuration level for the virtual server. For example, if server R1 has two application ports, 80 and 8080, enter the following commands to rebind the ports to virtual server VIP1. This example assumes that the VIP uses two real servers (R1 and R2) for the application ports.

```
ServerIron(config-vs-VIP1)# bind http R1 http R2 http
ServerIron(config-vs-VIP1)# bind 8080 R1 8080 R2 8080
```

## Keepalive Health Check State

When you configure a port profile for an application port, the keepalive health check for that port is enabled automatically. You also can enable or disable the keepalive health check for an application port on a specific real server, without affecting the global setting for the health check.

### USING THE CLI

To enable the keepalive health check, enter commands such as the following:

```
ServerIron(config)# server real Auto_Plooker 192.168.2.69
ServerIron(config-rs-Auto_Plooker)# port http keepalive
```

To disable the keepalive health check, enter commands such as the following:

```
ServerIron(config)# server real Auto_Plooker 192.168.2.69
ServerIron(config-rs-Auto_Plooker)# no port http keepalive
```

**Syntax:** [no] port <tcp/udp-port> keepalive

### USING THE WEB MANAGEMENT INTERFACE

1. Select the Real Server Port link from the bottom of the General SLB panel or another SLB panel.
2. Click the Modify button for the application port to be modified. The Real Server Port panel is displayed.
3. Click the Keep Alive checkbox. If the box contains a checkmark, the keepalive health check is enabled. If the box does not contain a checkmark, the keepalive health check is disabled.
4. Select the Modify button to save the change to the running-config.
5. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Connection Rate Limiting

Connection Rate Limiting (CRL) enables you to limit the connection rate to a real server for the following:

- All TCP traffic
- All UDP traffic
- Individual TCP or UDP ports

The ServerIron limits the number of new TCP, UDP, or individual port connections per second to the number you specify.

The ServerIron increments the connection counter for real server connections only after the ServerIron selects a server for the connection. If the ServerIron cannot serve a client request because a real server, cache, or firewall already has the maximum number of connections for the current second for the requested port, the ServerIron tries another server. If there are no servers available, the ServerIron sends a TCP RST to the client.

If you configure a limit for TCP or UDP and also for an individual application port, the ServerIron uses the lower limit. For example, if you limit new TCP connections to a real server to 1000 per second and also limit new HTTP connections to 600 per second, the ServerIron limits connections to TCP port HTTP to 600 per second.

---

**NOTE:** The ServerIron counts only the new connections that remain in effect at the end of the one second interval. If a connection is opened and terminated within the interval, the ServerIron does not include the connection in the total for the server.

---

---

**NOTE:** The connection limit you specify is enforced on an individual WSM CPU basis. Thus, each WSM CPU allows up to the number of connections you specify. For example, if you specify a maximum TCP connection rate of 800 connections per second, each WSM CPU allows up to 800 TCP connections per second, for a total of 2400 TCP connections per second.

---

To limit the number of new TCP and UDP connections a real server can receive each second, enter commands such as the following:

```
ServerIron(config)# server real RS1 1.2.3.4
ServerIron(config-rs-RS1)# max-tcp-conn-rate 1000
ServerIron(config-rs-RS1)# max-udp-conn-rate 800
```

The first command limits new TCP connections to the real server to 1000 per second. The second command limits the rate of new UDP connections to the real server to 800 per second.

**Syntax:** max-tcp-conn-rate <num>

**Syntax:** max-udp-conn-rate <num>

The <num> parameter specifies the maximum number of connections per second. There is no default.

To limit the rate of new connections for a specific application port, enter commands such as the following:

```
ServerIron(config-rs-RS1)# port http
ServerIron(config-rs-RS1)# port http max-tcp-conn-rate 600
```

These commands add port HTTP (80) to the real server and limit the rate of new connections to the port to 600.

**Syntax:** port <TCP/UDP-portnum> max-tcp-conn-rate <num>

**Syntax:** port <TCP/UDP-portnum> max-udp-conn-rate <num>

The **port** <TCP/UDP-portnum> parameter specifies the application port.

The <num> parameter specifies the maximum number of connections per second.

## Layer 7 Health Check Parameters

See “Customizing Layer 7 Health Checks” on page 12-31.

## Configuring Virtual Server Parameters

For basic virtual server (VIP) configuration, you need to specify a name and the virtual server’s IP address (the VIP), add the application ports that you want to load balance, then bind the VIP to the real servers based on the application ports. The following sections describe more advanced virtual server options.

### Application Ports and Bindings

You can add application ports to a virtual server and bind the virtual server to real servers when you first define the virtual server or later after you have defined the server. You can use the CLI or the Web management interface. See “Binding Virtual and Real Servers” on page 6-20.

#### USING THE CLI

To add an application port to a virtual server, enter commands such as the following:

```
ServerIron(config-rs-web4)# server virtual www.altergo.com 207.95.55.1
ServerIron(config-vs-www.altergo.com)# port http
```

**Syntax:** [no] port <tcp/udp-port>

This command has additional, optional parameters. See “Configuring Virtual Server Application Port Parameters” on page 6-60.

To bind a real server to a virtual server, enter commands such as the following:

```
ServerIron(config-rs-web4)# server virtual www.altergo.com
ServerIron(config-vs-www.altergo.com)# bind http web1 http
ServerIron(config-vs-www.altergo.com)# bind http web2 http
ServerIron(config-vs-www.altergo.com)# bind http web3 http
ServerIron(config-vs-www.altergo.com)# bind http web4 http
```

**Syntax:** bind <tcp/udp-port-number> <real-server-name> <tcp/udp-port-number>

---

**NOTE:** For clarity, the bindings in the example above are shown as four separate entries. Alternatively, you can enter all the binding information as one command: **bind http web1 http web2 http web3 http web4 http**

---



### USING THE WEB MANAGEMENT INTERFACE

1. Select the [Virtual Server Port](#) link from the Virtual Server entry panel.
2. Click the Modify button next to the virtual server to be modified.
3. Select the TCP or UDP port (service) you want the virtual port to support from the TCP/UDP Port pulldown menu.

---

**NOTE:** You also can define your own port rather than selecting one from the Port pulldown menu by selecting the User Define button.

---

4. Select the Add button to assign the TCP/UDP port to the virtual server. If you are changing parameters for a port that you already assigned to the virtual server, select Modify.
5. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.
6. Select the [Bind](#) link from the SLB, Real Server, or Virtual Server configuration panel to display the Bind table or the Bind configuration panel. If the Bind table is shown, select the [Add Bind](#) link.
7. Select the desired virtual server name from the Virtual Server Name pulldown menu.
8. Select the desired virtual server TCP/UDP port from the Virtual TCP/UDP Port pulldown menu. If you want to bind all the ports on the server, select Default.
9. Select the desired real server name from the Real Server Name pulldown menu.
10. Select the desired real server TCP/UDP port from the Real TCP/UDP Port pulldown menu. If you want to bind all the ports on the server, select Default.
11. Select the Bind button to map the selected virtual server and service to the selected real server and TCP/UDP server port.
12. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Primary and Backup Servers

By default, the virtual server uses the locally attached real servers (added using the **server real-name** command) as the primary load-balancing servers and uses the remotely attached servers (added using the **server remote-name** command) as backups.

You can enable the virtual server to use the servers designated as backups only as backups, and use the other servers as the primary load-balancing servers.

---

**NOTE:** This section does not apply to software release 07.1.x.

---

### Configuring Primary and Backup Servers

To configure primary and backup servers:

- Edit the configuration of each backup real server to designate the server as a backup. See "Backup Server" on page 6-38.

---

**NOTE:** You do not need to designate the primary servers. The ServerIron assumes that all servers you do not designate as primary servers are backup servers.

---

- Enable use of the primary and backup servers in individual VIPs on individual application ports. Only the VIPs and application ports for which you enable the feature use it. The other application ports within the VIP, and the other VIPs, use the locally-attached servers (configured using the **server real-name** command) as their primary servers and the remotely-attached servers (configured using the **server remote-name** command) as their backup servers.

Optionally, configure the individual applications on the VIPs to continue using the backup servers following a failover, instead of returning to the primary servers.

#### **Enabling a VIP to Use the Primary and Backup Servers**

To enable a VIP to use the servers designated as backups only as backups, and use the other servers as the primary load-balancing servers, enter the following command at the configuration level for the VIP:

```
ServerIron(config-vs-VIP1)# port http lb-pri-servers
```

This command enables VIP1 to use the backup and primary servers for application port HTTP.

To configure the VIP and application port to continue using the backup servers even after the primary servers become available again, use the backup-stay-active parameter, as in the following example:

```
ServerIron(config-vs-VIP1)# port http lb-pri-servers backup-stay-active
```

**Syntax:** [no] port <tcp/udp-port> lb-pri-servers [backup-stay-active]

For a complete CLI example, see “Backup Server” on page 6-38.

## **Host Range**

If you want to use the Unlimited VIP feature to load balance a large set of contiguous IP addresses, define a host range on the real servers and on the virtual server. Defining a host range simplifies configuration by allowing you to enter a single command or Web option for the whole range of addresses instead of entering information for each address individually.

---

**NOTE:** You also must configure the same size host range on each of the real servers.

---

For a complete configuration example, see “Web Hosting with Unlimited Virtual IP Addresses” on page 6-101.

#### **USING THE CLI**

Enter commands such as the following to configure a host range on a virtual server.

```
ServerIron(config)# server virtual-name v1 209.157.22.6
ServerIron(config-vs-v1)# host-range 20
```

**Syntax:** [no] host-range <range>

#### **USING THE WEB MANAGEMENT INTERFACE**

1. Select the Virtual Server link from the bottom of the General SLB panel or another SLB panel.
2. Click Modify next to the row of information about the virtual server you want to modify. If you are adding a new virtual server, select the Add Virtual Server link instead.
3. Edit the number in the Host Range field to specify how many contiguous IP addresses you want to configure based on the virtual server's IP address.
4. Select the Add or Modify button to assign the port or the changes to the port to the server.
5. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## **HTTP Redirect**

If you have configured some of the real servers as remote servers, you might also want to enable HTTP redirect on those servers. HTTP redirect causes the real servers to respond directly to clients instead of sending the response back through the ServerIron.

For a complete configuration example, see “Using HTTP Redirect with Geographically-Distributed Servers” on page 6-113.

#### **USING THE CLI**

To enable HTTP redirect on a virtual server, enter commands such as the following:

```
ServerIron(config)# server virtual-name VIP 209.157.22.88
```

```
ServerIron(config-vs-VIP1)# port http
ServerIron(config-vs-VIP1)# bind http r1 80 r2 80 r3 80
ServerIron(config-vs-VIP1)# httpredirect
```

**Syntax:** [no] httpredirect

#### USING THE WEB MANAGEMENT INTERFACE

1. Select the [Virtual Server](#) link from the bottom of the General SLB panel or another SLB panel.
2. Click Modify next to the row of information about the virtual server you want to modify. If you are adding a new virtual server, select the [Add Virtual Server](#) link instead.
3. Select the HTTP Redirect checkbox to place a checkmark in the box.
4. Select the Add or Modify button to assign the port or the changes to the port to the server.
5. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Load Balancing Method (Predictor)

You can override the globally configured load balancing method for an individual virtual server. The methods you can use are the same as the ones you can configure globally.

### Changing the Load Balancing Method

To change the load balancing method for an individual virtual server, use one of the following methods.

---

**NOTE:** If you use the server response time method, you also can modify the smooth factor on individual application ports. See "Smooth Factor" on page 6-62.

---

#### USING THE CLI

To change the load balancing method on an individual virtual server, enter commands such as the following:

```
ServerIron(config)# server virtual www.plookme.com 207.95.5.1
ServerIron(config-vs-www.plookme.com)# predictor response-time
```

**Syntax:** [no] predictor least-conn | least-local-conn | least-local-sess | response-time | round-robin | weighted

#### USING THE WEB MANAGEMENT INTERFACE

To define the virtual server name and address, enter the following:

1. Select the [Virtual Server](#) link from the SLB, Real Server, or Virtual Server entry panel.
2. Select Modify next to the virtual server.
3. Select the desired load balancing predictor. If Default is selected, the globally configured predictor is used.
4. Select the Modify button to assign the changes to the running-config.
5. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Symmetric SLB Priority

If you are configuring a pair of ServerIrons to provide redundancy for individual VIPs, you must specify an SLB priority on each ServerIron for each of the VIPs. The ServerIron with the higher priority for a given VIP is the default active ServerIron for that VIP. The other ServerIron is the default standby for the VIP.

For a configuration example and more information, see "Using Symmetric Server Load Balancing" on page 7-1.

#### USING THE CLI

To specify the priority, enter a command such as the following:

```
ServerIron(config)# server virtual-name noi-is-cool 1.2.3.4
ServerIron(config-vs-noi-is-cool)# sym-priority 254
```

**Syntax:** [no] sym-priority <num>

You can specify from 0 – 255. If you specify 0, the priority setting is removed.

---

**NOTE:** Foundry recommends that you specify 2 (instead of 1) as a low priority or 254 (instead of 255) as a high priority. This way, you can easily force failover of the high priority ServerIron to the low priority ServerIron by changing the priority on just one of the ServerIrons. For example, you can force a failover by changing the priority on the high priority ServerIron from 254 to 1. Since the priority on the low priority ServerIron is 2, the low priority ServerIron takes over for the VIP. Likewise, you can force the low priority ServerIron to take over by changing its priority to 255, since the priority on the high priority ServerIron is only 254.

---

#### USING THE WEB MANAGEMENT INTERFACE

1. Select the Virtual Server link from the SLB, Real Server, or Virtual Server entry panel.
2. Select Modify next to the virtual server.
3. Enter the priority in the Symmetric Priority field. You can specify from 0 – 255. If you specify 0, the priority setting is removed.
4. Select the Modify button to assign the changes to the running-config.
5. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Track Ports

You can configure the ServerIron to send all client requests for a specific set of TCP/UDP ports to the same real server as a "primary" TCP/UDP port grouped with the other ports. You can group a primary TCP/UDP port with up to four additional TCP/UDP ports. After the ServerIron sends a client request for the primary port to a real server, subsequent requests from the client for ports grouped with the primary port go to the same real server. See "TCP/UDP Application Groups" on page 6-104 for an example of application grouping.

---

**NOTE:** You must configure all the grouped ports to be "sticky".

---

---

**NOTE:** If a client requests one of the ports that follows the primary port before requesting the primary port itself, the ServerIron does not make the connection sticky. Only after the client requests the primary port does the ServerIron make subsequent requests from the client for that port or ports that track the primary port sticky.

---

---

**NOTE:** For servers that use passive FTP, configure the FTP ports to be both sticky and concurrent.

---

For a configuration example and more information, see "TCP/UDP Application Groups" on page 6-104.

#### USING THE CLI

To configure a TCP/UDP application group, enter the following commands. These commands configure HTTP (port 80), Telnet (port 23), and TFTP (port 69) to be sticky.

```
ServerIron(config)# server virtual-name v1 209.157.22.1
ServerIron(config-vs-v1)# port 80 sticky
ServerIron(config-vs-v1)# port 23 sticky
ServerIron(config-vs-v1)# port 69 sticky
ServerIron(config-vs-v1)# track 80 23 69
ServerIron(config-vs-v1)# bind 80 r1 80 r2 80
ServerIron(config-vs-v1)# bind 23 r1 23 r2 23
ServerIron(config-vs-v1)# bind 69 r1 69 r2 69
```

**Syntax:** [no] track <primary-port> <TCP/UDP-port> [<TCP/UDP-port> <TCP/UDP-port> <TCP/UDP-port>]

#### USING THE WEB MANAGEMENT INTERFACE

To configure an application port to be sticky:

1. Select the [Virtual Server Port](#) link from the bottom of the General SLB panel or another SLB panel. A list of the configured virtual server ports is displayed. Look in the Sticky column for the port.
  - If the column says “Yes”, then the port is sticky. Go to the next procedure.
  - If the column says “No”, you need to make the port sticky. Go to the next step.
2. Select Modify next to the row that describes the port. The Virtual Server Port panel for that port is displayed.
3. Select Sticky to enable the Sticky feature for the port.
4. Select Modify to assign the change.

To configure the application group:

1. Select the [Track](#) link to display the Track panel.
2. Select the primary port from the Primary Virtual Server Port field. The primary port is the port the other ports in the group will track.
3. Enter the numbers of the secondary ports in the Secondary Virtual Server Port List field. Separate each port with a blank space.

---

**NOTE:** The ports must already be individually defined on the virtual server.

---

4. Click Add to add the port to the application group.
5. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Track Port Group

A track port group is similar to track ports. The ServerIron sends a client's request for one of the ports to the same real server the ServerIron selected for the same client's request for another application port. The features differ in the following way:

- In a track port configuration, the tracking applies only to the primary port, which is the first port in the list of track ports. If the client requests one of the other applications (one of the applications that is tracking the primary application) first, the ServerIron track feature does not apply.
- In a track port group, the ServerIron sends a client's requests for any of the applications in the group to the same real server, regardless of which application the client requests first.

For a configuration example and more information, see “TCP/UDP Application Groups” on page 6-104.

To configure a track port group, use either of the following methods.

### USING THE CLI

The following commands illustrate the track group function:

```
ServerIron(config)# server virtual-name v1 209.157.22.1
ServerIron(config-vs-v1)# port 80 sticky
ServerIron(config-vs-v1)# port 69 sticky
ServerIron(config-vs-v1)# port 23 sticky
ServerIron(config-vs-v1)# track-group 80 69 23
ServerIron(config-vs-v1)# bind 80 r1 80 r2 80
ServerIron(config-vs-v1)# bind 23 r1 23 r2 23
ServerIron(config-vs-v1)# bind 69 r1 69 r2 69
ServerIron(config-vs-v1)# exit
```

**Syntax:** [no] track-group <TCP/UDP-port>...

In this example, the **track-group** command groups the HTTP port (80), Telnet port (23), and TFTP port (69) together. Whenever a client attempts to connect to a port within the group, the ServerIron ensures all ports in the group are active before granting the connection.

The **sticky** parameter makes the TCP/UDP ports sticky. The sticky parameter must be set for all ports in the group.

After the ServerIron sends a client to a real server for any of these three ports, subsequent requests from that client for the HTTP, TFTP, or Telnet port go to the same real server. Up to eight ports can be grouped together using the track group function. A port can be part of only one group. The **track-group** and **track** commands for a port are mutually exclusive.

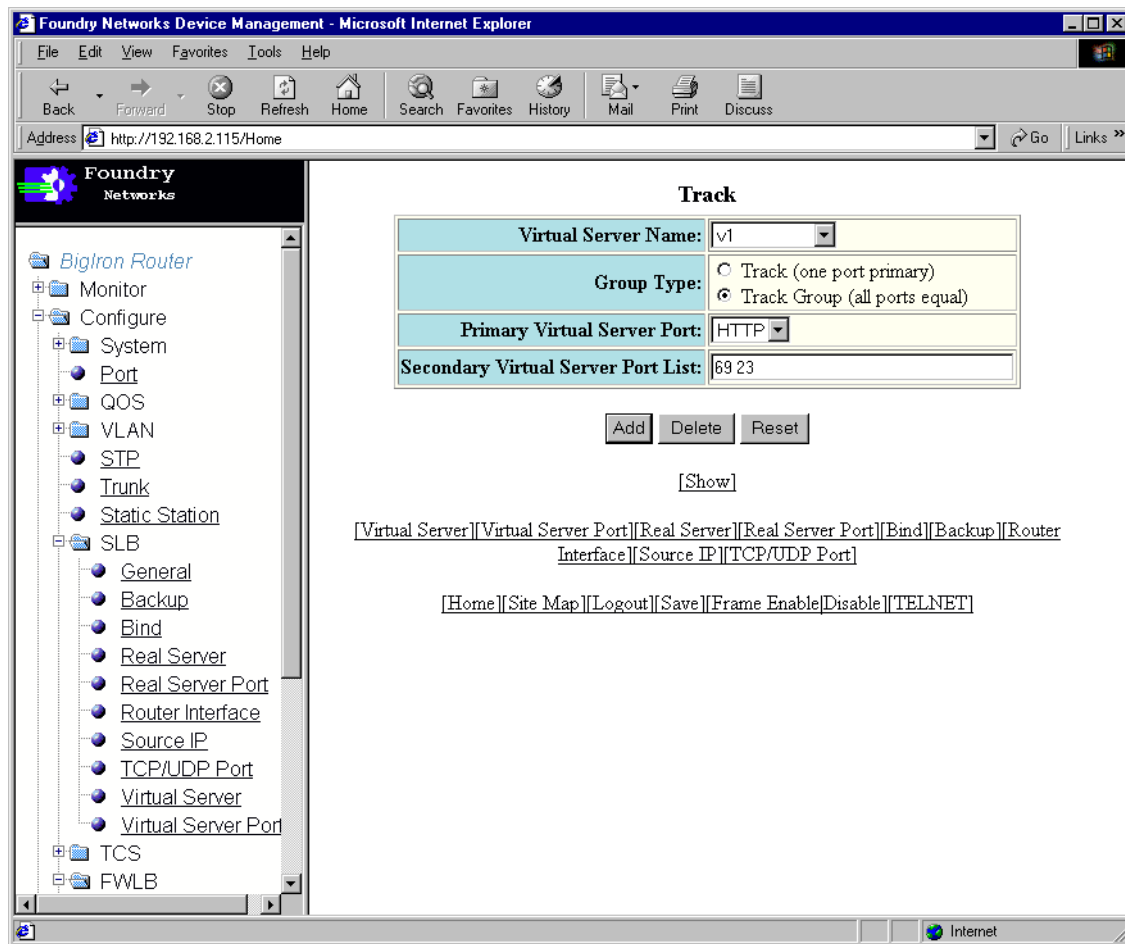
### USING THE WEB MANAGEMENT INTERFACE

To configure an application port to be sticky:

1. Select the Virtual Server Port link from the bottom of the General SLB panel or another SLB panel. A list of the configured virtual server ports is displayed. Look in the Sticky column for the port.
  - If the column says “Yes”, then the port is sticky. Go to the next procedure.
  - If the column says “No”, you need to make the port sticky. Go to the next step.
2. Select Modify next to the row that describes the port. The Virtual Server Port panel for that port is displayed.
3. Select Sticky to enable the Sticky feature for the port.
4. Select Modify to assign the change.

To configure the track group:

1. Select the Track link to display the Track panel. The following panel is displayed:



2. For the Group Type, select the button next to Track Group (all ports equal).

3. Select a port from the Primary Virtual Server Port field. In a track group, the primary port and the secondary ports are treated equally as a group.
4. Enter the numbers of the secondary ports in the Secondary Virtual Server Port List field. Separate each port with a blank space.

---

**NOTE:** The ports must already be individually defined on the virtual server.

---

5. Click Add to add the ports to the group.
6. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Enabling Server Cluster Support

---

**NOTE:** This section applies only to the ServerIron 400 and ServerIron 800.

---

In some configurations, such as those that use a cluster of servers for an application, you might want to configure the ServerIron to stop sending traffic for established connections to a server when the requested application is down on the server. For example, this feature is useful in an NFS configuration.

When you enable this feature, the ServerIron does one of the following in addition to redirecting future requests away from the real server:

- UDP – For an unavailable UDP application, the ServerIron terminates the connection.
- TCP – For an unavailable TCP application, the ServerIron resets the connection.

---

**NOTE:** The ServerIron always redirects new connections to real servers on which the requested application ports are available. The command in this section applies only to connections that are already established when the application fails.

---

To configure the ServerIron to stop sending requests for an established connection to a real server for an application that is down on the server, enter the following command at the configuration level for the VIP:

```
ServerIron(config-vs-VIP1)# port 80 reset-on-port-fail
```

This command configures the ServerIron to stop sending traffic on existing HTTP connections to a real server bound to VIP1 if the HTTP application has failed on the server. The ServerIron instead terminates the connection (if UDP) or resets the connection (if TCP).

**Syntax:** [no] port <tcp/udp-portnum> reset-port-on-fail

## Fast Aging for UDP Sessions

When fast aging for UDP sessions is configured, a client request causes the ServerIron to add an entry to its session table; when a response is detected, the ServerIron immediately deletes the session table entry.

---

**NOTE:** This section applies only to the ServerIron 400 or ServerIron 800 running software release 07.2.20 or later.

---



---

**NOTE:** Fast aging is the default behavior for the well-known DNS and RADIUS ports. To change DNS or RADIUS to use the UDP age timer instead, see "Normal UDP Aging for DNS and RADIUS" on page 6-60.

---

When this feature is configured, if the ServerIron detects a server response to a client request, and the response is not fragmented, the session table entry is deleted immediately. If the response is fragmented, the ServerIron waits for the last fragment to arrive, forwards it to the client, and then sends the session to the delete queue. The session stays in the delete queue for 8 seconds by default before being deleted. You can change the amount of time the session stays in the delete queue to between 1 – 40 seconds.

To activate this feature for port 1234:

```
ServerIron(config)# server virtual vs1 192.168.1.2
ServerIron(config-vs-vs1)# port 1234 udp-fast-age
```

**Syntax:** port <UDP-portnum> udp-fast-age

To set the amount of time sessions for ports configured with the **udp-fast-age** command stay in the delete queue before being deleted:

```
ServerIron(config)# server msl 2
```

**Syntax:** server msl <secs>

The <secs> parameter can be from 1 – 40 seconds.

## Normal UDP Aging for DNS and RADIUS

By default, the ServerIron immediately deletes a UDP DNS or RADIUS session table entry when the ServerIron receives a reply for the application from a real server. You can configure the ServerIron to instead age DNS or RADIUS sessions out normally using the UDP age timer.

---

**NOTE:** This section applies only to the ServerIron 400 or ServerIron 800 running software release 07.2.25 or later.

---

To age DNS or RADIUS sessions using the UDP age timer, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config-vs-VIP1)# port dns udp-normal-age
```

**Syntax:** [no] port dns | radius udp-normal-age

## Transparent VIP

Transparent VIP allows you to configure a ServerIron to transparently load balance a VIP, without owning the VIP address. This feature is useful when you want to configure multiple ServerIrons to load balance for the same VIP. For examples and configuration information, see “Configuring Transparent VIPs and Stateless SLB” on page 8-1.

### USING THE CLI

To configure an individual virtual server for the transparent VIP feature, enter a command such as the following:

```
ServerIron(config-vs-TransVIP)# transparent-vip
```

**Syntax:** [no] transparent-vip

### USING THE WEB MANAGEMENT INTERFACE

You cannot configure this feature using the Web management interface.

## Configuring Virtual Server Application Port Parameters

The following sections describe how to modify parameters for application ports on virtual servers.

### Port State

Application ports are enabled by default. To disable an application port on a real server, use either of the following methods.

#### USING THE CLI

```
ServerIron(config)# server virtual Zoot_Allures 1.2.3.4
ServerIron(config-vs-Zoot_Allures)# port http disable
```

**Syntax:** [no] port <tcp/udp-port> disable

To re-enable a port, enter commands such as the following:

```
ServerIron(config)# server virtual Zoot_Allures 1.2.3.4
```



```
ServerIron(config-vs-Zoot_Allures)# no port http disable
```

### USING THE WEB MANAGEMENT INTERFACE

1. Select the [Virtual Server Port](#) link from the bottom of the General SLB panel or another SLB panel.
2. Click the Modify button for the application port to be modified. The Virtual Server Port panel is displayed.
3. Click Enable or Disable next to Status.
4. Select the Modify button to save the change to the running-config.
5. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Sticky

By default, the ServerIron sends a client's request to the next available real server based on the load balancing method. This is true regardless of whether the client has already sent a request for the same application. If you want the ServerIron to send all of a client's requests for a given application to the same real server during a client's session with the server, configure the application port to be sticky.

The port tracking and port group features require the application ports to be sticky.

---

**NOTE:** For servers that use passive FTP, configure the FTP ports to be both sticky and concurrent.

---

For a configuration example and more information, see "TCP/UDP Application Groups" on page 6-104.

### USING THE CLI

To configure a TCP/UDP application group, enter the following commands. These commands configure HTTP (port 80), Telnet (port 23), and TFTP (port 69) to be sticky. In addition, the Telnet and TFTP ports are configured to track the HTTP port.

```
ServerIron(config)# server virtual-name v1 209.157.22.1
ServerIron(config-vs-v1)# port 80 sticky
```

**Syntax:** [no] port <tcp/udp-port> sticky

### USING THE WEB MANAGEMENT INTERFACE

To configure an application port to be sticky:

1. Select the [Virtual Server Port](#) link from the bottom of the General SLB panel or another SLB panel. A list of the configured virtual server ports is displayed. Look in the Sticky column for the port.
  - If the column says "Yes", then the port is already sticky.
  - If the column says "No", you need to make the port sticky. Go to the next step.
2. Select Modify next to the row that describes the port. The Virtual Server Port panel for that port is displayed.
3. Select Sticky to enable the feature for the port.
4. Select Modify to assign the change.

## Concurrent

The concurrent feature allows a client to have sessions on different application ports on the same real server at the same time. When you enable an application port to be concurrent, the real server can open additional ("concurrent") TCP/UDP sessions with the client using arbitrary TCP/UDP port numbers.

Although the concurrent connections attribute is similar to application groups, application groups apply to specific TCP/UDP ports that you configure on the virtual server.

---

**NOTE:** For servers that use passive FTP, configure the FTP ports to be both sticky and concurrent.

---

### USING THE CLI

```
ServerIron(config)# server virtual-name v1 209.157.22.1
ServerIron(config-vs-v1)# port 80 concurrent
```

**Syntax:** [no] port <tcp/udp-port> concurrent

### USING THE WEB MANAGEMENT INTERFACE

To configure an application port for concurrency:

1. Select the [Virtual Server Port](#) link from the bottom of the General SLB panel or another SLB panel. A list of the configured virtual server ports is displayed. Look in the Concurrent column for the port.
  - If the column says “Yes”, then the port is already configured for concurrency.
  - If the column says “No”, you need to enable the port for concurrency. Go to the next step.
2. Select Modify next to the row that describes the port. The Virtual Server Port panel for that port is displayed.
3. Select Concurrent to enable the feature for the port.
4. Select Modify to assign the change.

## SwitchBack (DSR)

The SwitchBack feature allows a real server to use a return path that does not pass through the ServerIron.

Normally, the ServerIron can perform health checks on an application port only when server replies from that port pass back through the ServerIron. If the ServerIron does not see the real server’s responses to client requests, the ServerIron concludes that the application or the entire server is down and stops sending client requests to that server.

When you enable an application port for SwitchBack, the ServerIron can still perform health checks on the application by sending the health checks to the loopback address you configure on the real server.

For a configuration example and more information, see “Configuring Symmetric SLB and SwitchBack” on page 7-1.

### USING THE CLI

```
ServerIron(config)# server virtual-name v1 209.157.22.1
ServerIron(config-vs-v1)# port 80 dsr
```

**Syntax:** [no] port <tcp/udp-port> dsr

### USING THE WEB MANAGEMENT INTERFACE

To configure an application port for concurrency:

1. Select the [Virtual Server Port](#) link from the bottom of the General SLB panel or another SLB panel. A list of the configured virtual server ports is displayed. Look in the DSR column for the port.
  - If the column says “Yes”, then the port is already configured for SwitchBack.
  - If the column says “No”, you need to enable the port for SwitchBack. Go to the next step.
2. Select Modify next to the row that describes the port. The Virtual Server Port panel for that port is displayed.
3. Select DSR to enable the feature for the port.
4. Select Modify to assign the change.

## Smooth Factor

This section applies to the server response time load balancing method.

The ServerIron calculates the server response time value for a real server by regularly collecting response time samples, then using a calculation to smooth the values of the samples and derive a single response time value for

the real server. The ServerIron collects the samples around once every 100 milliseconds (about 10 times a second). The sampling rate can vary slightly depending on the processing the ServerIron is performing.

To smooth the samples out, the ServerIron uses the following calculation:

$$R = ((S * \text{previous\_R}) + ((100 - S) * \text{new\_R})) / 100$$

where:

R = Response time

S = Smooth factor, which is configurable and can be from 1 – 99. The default is 90. A large value gives the previous response time more weight than the new response time. A small value gives the new response time more weight than the previous response time.

For example, if a given real server's previous response time value was 2 milliseconds, and a new measurement also results in 2 milliseconds, the calculated server response time using the default smooth factor is as follows:

$$R = ((90 * 2) + ((100 - 90) * 2)) / 100$$

$$R = 180 + 20 / 100$$

$$R = 200 / 100$$

$$R = 2$$

If the real server's response time slows due to processing for additional connections, the slower response time affects the Server Response Time calculation for the server. For example, if the next server response time sample is 5 milliseconds instead of 2, the Server Response Time calculation is as follows:

$$R = ((90 * 2) + ((100 - 90) * 5)) / 100$$

$$R = 180 + 50 / 100$$

$$R = 230 / 100$$

$$R = 2.3$$

Since the real server's response time has slowed by two and a half times, the server's response time calculation biases the ServerIron away from selecting that server for new connections.

You can affect the degree of difference in subsequent response time weights by changing the smooth factor (S). For example, if you change the smooth factor from 90 (the default) to 50, the calculations shown above have the following results:

Here is the calculation when the previous and new response times are 2 milliseconds:

$$R = ((50 * 2) + ((100 - 50) * 2)) / 100$$

$$R = 100 + 100 / 100$$

$$R = 200 / 100$$

$$R = 2$$

Here is the calculation if the server's next response time is 5 milliseconds.

$$R = ((50 * 2) + ((100 - 50) * 5)) / 100$$

$$R = 100 + 250 / 100$$

$$R = 350 / 100$$

$$R = 3.5$$

Notice that the differences between the first and second samples are much greater when the smooth factor is 50 than when the smooth factor is 90. A large value gives the previous response time more weight than the new response time. A small value gives the new response time more weight than the previous response time.

You can change the smooth factor on an individual virtual server basis and on an individual application port basis.

- If you change the smooth factor for a virtual server, the change affects all Server Response Time calculations

for the real servers bound to the virtual server.

- If you change the smooth factor for an application port, the change affects Server Response Time calculations only for port bindings that use that application port.

#### USING THE CLI

To change the smooth factor for a virtual server, enter a command such as the following at the configuration level for the virtual server:

```
ServerIron(config-vs-Joes_Garage)# port 80 smooth-factor 50
```

**Syntax:** [no] smooth-factor <num>

The <num> parameter specifies the smooth factor value the server response time calculation uses. You can specify a number from 1 – 99. The default is 90.

#### USING THE WEB MANAGEMENT INTERFACE

You cannot configure this parameter using the Web management interface.

## Stateless

By default, the ServerIron creates session table entries for sessions between clients and applications on real servers. The ServerIron uses the session table entries to maintain state information for the sessions. The ServerIron uses the state information for features such as health checking and session failover in hot-standby configurations.

You can configure individual application ports to be stateless. The ServerIron does not maintain state information for a stateless port. Making a port stateless is useful when you want to conserve session table resources or when you have configured the VIP to be transparent.

For examples and configuration information, see “Configuring Transparent VIPs and Stateless SLB” on page 8-1.

#### USING THE CLI

To configure an application port to be stateless, enable the stateless parameter on the port in the virtual server. Here is an example:

```
ServerIron(config)# server real R1 10.10.10.1
ServerIron(config-rs-R1)# port http
ServerIron(config-rs-R1)# exit
ServerIron(config)# server real R2 10.10.11.1
ServerIron(config-rs-R2)# port http
ServerIron(config-rs-R2)# exit
ServerIron(config)# server virtual StatelessHTTP 192.168.4.69
ServerIron(config-vs-StatelessHTTP)# port http stateless
ServerIron(config-vs-StatelessHTTP)# bind http R1 http
ServerIron(config-vs-StatelessHTTP)# bind http R2 http
```

**Syntax:** [no] port <tcp/udp-portnum> stateless

The <tcp/udp-portnum> parameter specifies the application port you want to make stateless.

---

**NOTE:** This software release supports stateless SLB only for TCP port 80 (HTTP).

---

#### USING THE WEB MANAGEMENT INTERFACE

You cannot configure this parameter using the Web management interface.

## Virtual Source

In a typical configuration, a client’s IP address remains the same throughout the client’s session with a virtual server. However, in some configurations where a proxy is used for the clients before the client traffic reaches the ServerIron, the client’s IP address can be different for each request. To configure session persistence in a proxy environment, configure a standard IP ACL containing the addresses, then use the **Virtual Source** feature.

When you configure the Virtual Source feature, the ServerIron sends all client traffic from a specified range of IP addresses to the same real server for the application ports you specify. To specify the IP addresses, configure a standard IP ACL. Use this command in configurations where a proxy device allocates IP addresses to client traffic before sending the traffic to the VIP. In some configurations, the proxy device assigns different IP addresses to traffic from the same client. Unless you configure the addresses to go to the same real server, the ServerIron might load balance the client traffic to different servers. This makes applications that require continued access to the same real server unusable.

#### USING THE CLI

To configure the Virtual Source feature, enter commands such as the following:

```
ServerIron(config)# access-list 1 permit 209.157.22.0
ServerIron(config)# server virtual fromproxy 1.1.1.1
ServerIron(config-vs-fromproxy)# port 80 sticky-acl 1
```

**Syntax:** [no] access-list <num> deny | permit <source-ip> | <hostname> <wildcard> [log]

or

**Syntax:** [no] access-list <num> deny | permit <source-ip>/<mask-bits> | <hostname> [log]

**Syntax:** [no] port <tcp/udp-port> sticky-acl <num>

**NOTE:** This feature is different from the sticky feature, which you can associate with ports on the virtual server level. The sticky attribute ensures that subsequent packets from the same client during the same TCP session go to the same real server. In this case, the ServerIron knows the packets are from the same client based on the source IP address. When a proxy is used, subsequent packets from the same client can have different IP addresses.

For standard IP ACL configuration information, see the “Configuring Standard ACLs” section in the “Using Access Control Lists (ACLs)” chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide*.

#### USING THE WEB MANAGEMENT INTERFACE

You cannot configure this feature using the Web management interface.

## Translation

By default, the ServerIron translates the application port number requested by the client into the application port number you specify on the virtual server when you bind it to the real server. For example, if you bind port 80 on a virtual server to port 8080 on a real server, the ServerIron translates the application port in the client's request from port 80 into 8080 before forwarding the request to a real server.

A few ServerIron configurations require that you disable translation for an application port. For example, if you want to bind multiple virtual IP addresses to the same real server, you must disable port translation for all but one of the virtual IP addresses, then bind the virtual IP addresses to an alias port for the application. Disabling port translation enables the virtual IP addresses to use the same actual port number on the real server while the ServerIron collects and displays separate statistics for the alias port number associated with each virtual IP address.

For a complete configuration example, see “Many-To-One TCP/UDP Port Binding” on page 6-98.

#### USING THE CLI

```
ServerIron(config)# server virtual-name v1 209.157.22.1
ServerIron(config-vs-v1)# no port 80 translate
```

**Syntax:** [no] port <tcp/udp-port> translate

#### USING THE WEB MANAGEMENT INTERFACE

To configure an application port for concurrency:

1. Select the [Virtual Server Port](#) link from the bottom of the General SLB panel or another SLB panel. A list of the configured virtual server ports is displayed. Look in the Translate column for the port.

- If the column says “Yes”, then port translation is enabled.
  - If the column says “No”, then port translation is disabled.
2. Select Modify next to the row that describes the port. The Virtual Server Port panel for that port is displayed.
  3. Select Translate to disable or re-enable the feature for the port. If the checkbox does not contain a checkmark, translation is disabled.
  4. Select Modify to assign the change.

## Enhanced SSL Accelerator Support

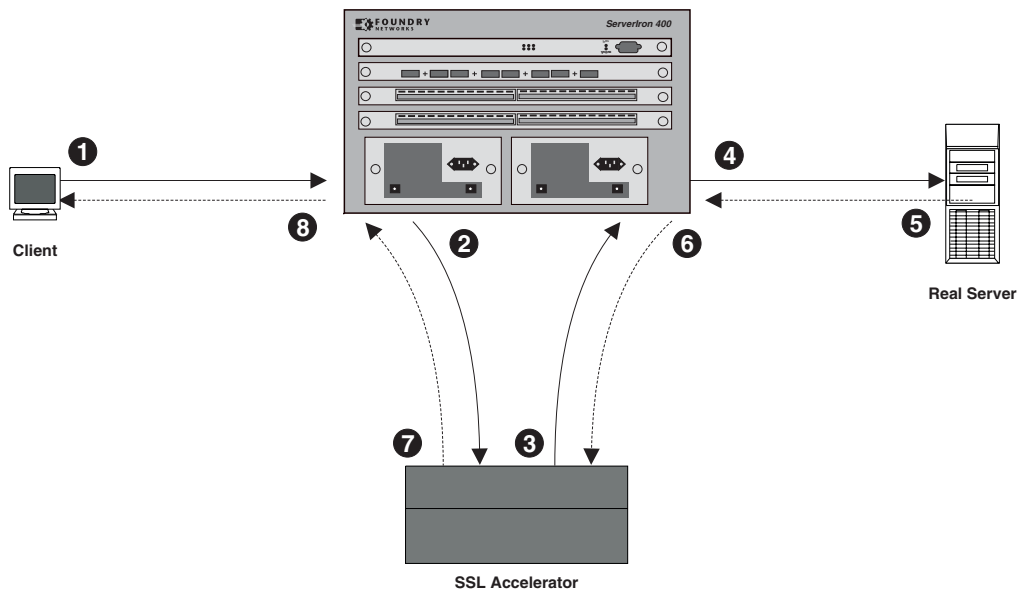
The ServerIron features enhanced support for SSL accelerators by allowing the ServerIron to send return traffic from a real server back to the SSL accelerator from which it was sent.

Normally, when the ServerIron supports SLB for some services and TCS for others, the cache server uses the original client's IP address as the source IP address for SLB traffic sent from the cache server to the ServerIron. When the ServerIron sends return traffic from the real server back to the client, it goes directly to the original client (bypassing the cache server).

However, some configurations (such as those using an SSL accelerator as a cache server) may require that traffic from a real server first go back to the cache server before going to the original client. Using a technique called **VIP spoofing**, the ServerIron, when it receives traffic from a real server on a specified port, forwards it not to the original client, but to the cache server where the SLB traffic was initiated.

The following diagram illustrates a configuration that uses VIP spoofing to direct SLB traffic from a real server to the SSL accelerator that originated the traffic.

**Figure 6.12 Using VIP spoofing with an SSL accelerator**



In this configuration, SSL traffic travels from the client to the real server as follows:

1. The client sends an SSL packet to a ServerIron VIP on port 443.
2. The ServerIron directs the packet to the SSL accelerator on port 443
3. The SSL accelerator sends the packet to the ServerIron on port 80.
4. The ServerIron directs the packet to the real server on port 80.
5. The real server sends a packet to the ServerIron on port 80.
6. The ServerIron sends packet to the SSL accelerator on port 80.

Normally, the ServerIron would send the packet directly back to the original client on port 80. However, with the VIP spoofing feature enabled, the ServerIron instead sends the packet to the cache server that initiated the traffic (in this case the SSL accelerator).

7. The SSL accelerator sends the packet back to the ServerIron on port 443.
8. The ServerIron sends the packet to the client on port 443.

To implement a configuration like the one in Figure 6.12, enter the following commands:

```
ServerIron(config)# server cache-name cs1 10.10.1.10
ServerIron(config-rs-cs1)# port ssl
ServerIron(config-rs-cs1)# port ssl no-health-check
ServerIron(config-rs-cs1)# port http
ServerIron(config-rs-cs1)# port http no-health-check
ServerIron(config-rs-cs1)# port http url "HEAD /"
ServerIron(config-rs-cs1)# exit

ServerIron(config)# server real rs1 10.10.1.40
ServerIron(config-rs-rs1)# port http
ServerIron(config-rs-rs1)# port http url "HEAD /"
ServerIron(config-rs-rs1)# exit

ServerIron(config)# server virtual vip1 10.10.1.100
ServerIron(config-vs-vip1)# port http
ServerIron(config-vs-vip1)# port http spoofing
ServerIron(config-vs-vip1)# port ssl
ServerIron(config-vs-vip1)# port ssl sticky
ServerIron(config-vs-vip1)# port ssl cache-enable
ServerIron(config-vs-vip1)# bind http rs1 http
ServerIron(config-vs-vip1)# exit

ServerIron(config)# server cache-group 1
ServerIron(config-tc-1)# cache-name cs1
ServerIron(config-tc-1)# exit

ServerIron(config)# ip address 10.10.1.1 255.255.255.0
ServerIron(config)# ip default-gateway 10.10.1.3
ServerIron(config)# ip policy 1 cache tcp 0 global
ServerIron(config)# ip policy 2 cache tcp ssl global
```

You can also configure the ServerIron so that the client's request to the VIP is translated to the real IP address of the cache server (that is, the SSL Accelerator) and then sent there. In this case, the port **ssl cache-enable** command is not used in the VIP's configuration. Instead, the cache server is bound to the SSL port on the VIP. In the example above, VIP vip1 would have the following configuration:

```
ServerIron(config)# server virtual vip1 10.10.1.100
ServerIron(config-vs-vip1)# port http
ServerIron(config-vs-vip1)# port http spoofing
ServerIron(config-vs-vip1)# port ssl
ServerIron(config-vs-vip1)# port ssl sticky
ServerIron(config-vs-vip1)# bind ssl cs1 ssl
ServerIron(config-vs-vip1)# bind http rs1 http
ServerIron(config-vs-vip1)# exit
```

**Syntax:** port http spoofing

## Configuring an IP Filter for a TCP/UDP Port

You can configure IP filters or IP Access Control Lists (ACLs) to explicitly permit or deny access to specific TCP/UDP ports. When you configure this type of filter, you specify the virtual IP address (VIP) as the destination address for the filter.

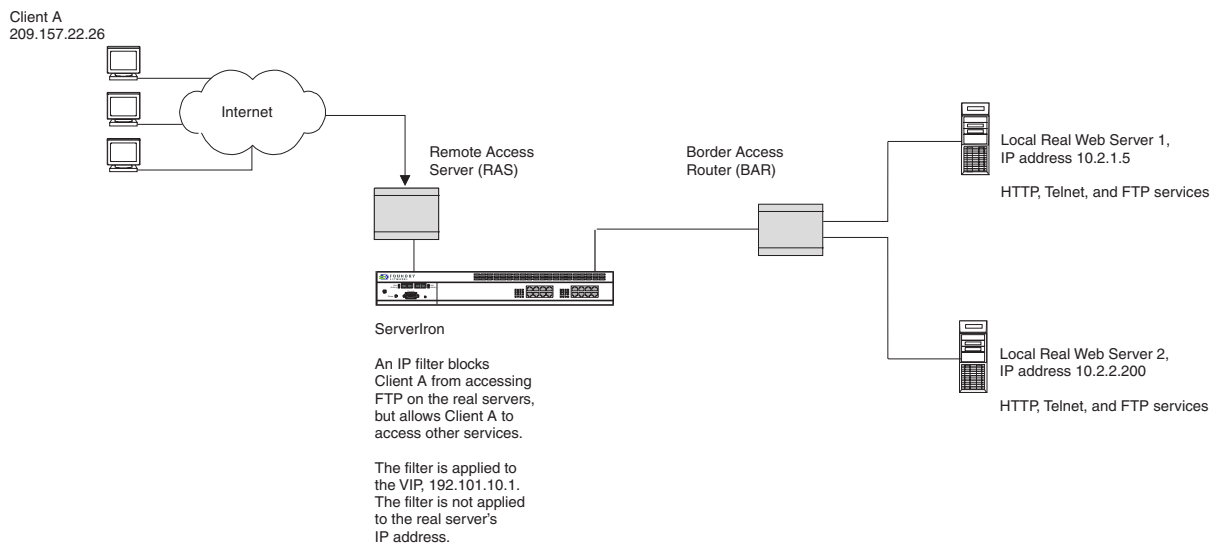
This section shows how to configure IP filters. For ACL configuration information, see the “Using Access Control Lists (ACLs)” chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide*.

**NOTE:** You cannot use Layer 2 filters to filter Layer 4 information. To filter Layer 4 information, use IP access policies. See the “Policies and Filters” appendix in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

Figure 6.13 shows an example of how you can use an IP filter in SLB. In this example, the administrator wants to block a specific client’s access to the FTP service on a VIP but permit access to the other services.

**NOTE:** When you configure the filter, specify the VIP as the destination address, not the real server’s IP address.

**Figure 6.13** IP filter used to block client access to a TCP/UDP port



To configure an IP filter for SLB, use one of the following methods.

#### USING THE CLI

To configure an IP filter to block 209.157.22.26 from accessing FTP on 192.101.10.1, enter the following command:

```
ServerIron(config)# ip filter 1 deny 209.157.22.26 255.255.255.0 192.101.10.1 255.255.255.0 tcp eq ftp
```

**Syntax:** ip filter <filter-id> permit | deny <src-ip-addr> | any <src-mask> | any <dst-ip-addr> | any <dst-mask> | any <protocol> [<established> <operator> <port range>]; items in brackets apply to TCP only.

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to System in the tree view to expand the list of system configuration option links.
4. Select the IP Filter link to display the IP Filter configuration panel.
5. Select the type of filter (action) to be defined: Deny or Permit. In this example, select Deny.
6. If you want to define the filter for a specific source address, enter the IP address of the source sub-net in the Source Address field. If you leave the default address 0.0.0.0 in this field, the filter is applied to all received traffic. In this example, specify 209.157.22.26.



7. If you entered an IP address in the previous step, enter the source mask in the Source Mask field. In this example, specify 255.255.255.0.
8. If you want to define the filter for a specific destination address, enter the IP address of the destination subnet in the Destination Address field. If you leave the default address 0.0.0.0 in this field, the filter is applied to all forwarded traffic. In this example, specify 192.101.10.1.
9. If you entered an IP address in the previous step, enter the destination mask in the Destination Mask field. In this example, specify 255.255.255.0.
10. Enter the protocol to be filtered. In this example, specify TCP.
11. Select the comparison operator, unless you want to use the default (Equal). In this example, use Equal.
12. Enter the TCP/UDP port number. In this example, enter 21, the well-known port number for an FTP control session.
13. Select the Add button to assign the change.
14. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Shutting Down a Real Server

The force shutdown feature (sometimes called the force delete feature) allows you to force termination of existing SLB connections. This feature assumes that you already have shut down a TCP/UDP service on the real server or you have shut down the real server itself.

There are several methods for shutting down a service or server. Each method has consequences, so choose the method that works best in your situation.

- Edit the real server configuration on the ServerIron to disable the TCP/UDP ports on the server. For example, to disable port 80 (HTTP), you can use the **port http disable** command at the real server level of the CLI. If you use this method, you do not need to re-define the real server to add the server back to SLB. However, you do need to re-enable the disabled TCP/UDP ports.
- Delete the real server from the ServerIron. This option immediately prevents new connections. The ServerIron allows existing connections to end normally or, if you have enabled the force shutdown option, the ServerIron ends all connections within two minutes. If you use this method, to re-add the real server to the ServerIron, you must redefine the real server, then rebind the real server to the appropriate VIP(s).
- Shut down the real server itself, rather than change definitions on the ServerIron. When the real server stops responding to health checks, the ServerIron removes the server from the SLB. This option is simple because it does not require any configuration changes on the ServerIron. However, this option immediately disconnects all users, whereas the above options allow the server or service to gracefully shut down (unless you use the force shutdown option).

## Viewing SLB Configuration Details and Statistics

You can view the following SLB configuration details and statistics:

- Global information – see “Displaying Global Configuration Information” on page 6-70
- Real server information – see “Displaying Real Server Information” on page 6-74
- Virtual server information – see “Displaying Virtual Server Information” on page 6-82
- Port-binding information – see “Displaying Port-Binding Information” on page 6-92
- Session information and statistics – see “Displaying Session Statistics” on page 6-93
- Traffic statistics – see “Displaying Traffic Statistics” on page 6-94
- Symmetric SLB information – see “Displaying Symmetric SLB Information” on page 7-14

## Displaying Global Configuration Information

You can display global Layer 4 configuration information using either of the following methods.

### USING THE CLI

To display global ServerIron configuration information, enter the following command at any level of the CLI:

```
ServerIron(config)# show server global

Server Load Balancing - global parameters
Predictor =          least-conn
Force-deletion =     0
Reassign-threshold = 20
Reassign-limit =     3
Ping-interval =      2
Ping-retries =       4
TCP-age =            30
UDP-age =            5
Sticky-age =         30
TCP-syn-limit =      65535
TCP-total conn =     4233
Unsuccessful conn =  0
ICMP-message = Disabled
```

This display shows the following information.

**Table 6.5: Global Layer 4 Configuration Information**

This Field...	Displays...
<b>Symmetric SLB Parameters</b>	
You also can display this information separately. See “Displaying Symmetric SLB Information” on page 7-14.	
Server Symmetric port	The ServerIron port number on which the ServerIron perceives other ServerIrons running Symmetric SLB.
Group_id	The Symmetric SLB group ID. The group ID is always 1 in the current release.
Candidate cnt	The number of ports on which the ServerIron perceives a partner ServerIron running Symmetric SLB.
Port	The TCP/UDP port for which Symmetric SLB is enabled.
Priority	The priority for the VIP.
No-rx	Information Foundry technical support can use to help resolve Symmetric SLB configuration issues.
<b>SLB Parameters</b>	

Table 6.5: Global Layer 4 Configuration Information (Continued)

This Field...	Displays...
Predictor	<p>The load balancing metric in effect on the ServerIron. The predictor can be one of the following:</p> <ul style="list-style-type: none"> <li>least-conn (least connections)</li> <li>least-sess (least sessions)</li> <li>response-time (server response time)</li> </ul> <p><b>Note:</b> This value also applies to the combined method of least connections and server response time weights.</p> <ul style="list-style-type: none"> <li>round-robin (round robin)</li> <li>weighted (weighted percentage)</li> <li>least-local-conn (least local connections)</li> <li>least-local-sess (least local sessions)</li> </ul> <p>The default is least-conn.</p> <p>You can assign these metrics on a global basis and an individual virtual server basis.</p> <p>For more information or to globally change the predictor, see “Load Balancing Method (Predictor)” on page 6-24.</p> <p>To locally change the predictor for a virtual server, see “Load Balancing Method (Predictor)” on page 6-55.</p>
Force-deletion	<p>The state of the force shutdown option. This option immediately shuts down a server or service instead of waiting for existing connections to end before shutting the server or service down. The state can be one of the following:</p> <ul style="list-style-type: none"> <li>0 – Disabled</li> <li>1 – Enabled</li> </ul> <p>See the section on shutting down a server in “Configuring Server Load Balancing” on page 6-1 or “Configuring Transparent Cache Switching” on page 10-1 for more information.</p>
Reassign-threshold	<p>The number of contiguous inbound TCP-SYN packets sent to the server that the server has not responded to.</p> <p>The TCP SYN-ACK counter increments only when acknowledgments are not received. Each time an expected TCP SYN-ACK is received, the counter is cleared.</p> <p>The default reassign threshold is 21 unacknowledged TCP SYN-ACKs. The value can be from 6 – 254. To change the reassign threshold, see “Modifying the Reassign Threshold” on page 12-29</p> <p><b>Note:</b> You can modify this parameter to help minimize vulnerability to TCP SYN attacks.</p>
Reassign-limit	<p>The number of missed TCP SYN packets the ServerIron will accept before moving an inbound connection attempt to another server.</p>
<b>Layer 3 Health Check Parameters</b>	

**Table 6.5: Global Layer 4 Configuration Information (Continued)**

This Field...	Displays...
Ping-interval	How often the ServerIron sends a Layer 3 IP ping to test the basic health and reachability of the real servers. When enabled, this parameter specifies the interval for the pings. To change the interval, see “Modifying the Ping Interval and Retries” on page 12-19.
Ping-retries	How many times the ServerIron resends a ping to a real server that is not responding. The default is 4 and can be from 2 – 10. To change this parameter, see “Modifying the Ping Interval and Retries” on page 12-19.  If the server still does not respond after the last retry, the ServerIron marks the server FAILED and removes it from the load balancing rotation.
<b>Global TCP/UDP Parameters</b>	
TCP-age	The number of minutes the ServerIron allows a TCP connection to remain unused before closing the connection. The default is 30 minutes. To change this parameter, see “Modifying the TCP Age” on page 12-59.  The value shown here is the global value. You can override this value for an individual TCP/UDP port by modifying its port profile. See “Overriding the Global TCP or UDP Age” on page 12-27.
UDP-age	The number of minutes the ServerIron allows a UDP connection to remain unused before closing the connection. The default is 5 minutes. To change this parameter, see “Modifying the UDP Age” on page 12-59.  The value shown here is the global value. You can override this value for an individual TCP/UDP port by modifying its port profile. See “Overriding the Global TCP or UDP Age” on page 12-27.
Sticky-age	The number of minutes a sticky server connection can remain inactive before aging out. The default is 5 minutes.
TCP-syn-limit	The maximum number of TCP SYN connections per second the ServerIron is allowed to send to the real server. The default is 65535 connections.  You can guard against TCP SYN attacks by changing this parameter to a lower value. See “TCP SYN Limit” on page 6-27.
<b>TCP Connections Counters</b>	
TCP-total conn	The total number of TCP connections the ServerIron is currently managing.
Unsuccessful conn	The number of client requests for a TCP port that the ServerIron could not fulfill because the server was busy or down, or the port was not configured on the server.
<b>ICMP Message Feature State</b>	

**Table 6.5: Global Layer 4 Configuration Information (Continued)**

This Field...	Displays...
ICMP-message	<p>The state of the ICMP message feature. The state can be one of the following:</p> <ul style="list-style-type: none"> <li>Disabled – The ServerIron does not send ICMP “Destination Unreachable” messages to a client that requests an HTTP port that is on a busy or down server or that is not configured on the server. This is the default.</li> <li>Enabled – The ServerIron does send ICMP “Destination Unreachable” messages to clients when the requested HTTP port is not available or not configured.</li> </ul> <p>To change the state of this feature, see “ICMP Unreachable Messages” on page 6-29.</p>

### USING THE WEB MANAGEMENT INTERFACE

To display general ServerIron configuration information, do one of the following:

- Select the General link from the list of statistics links at the bottom of the General panel or other SLB panels that have the links.
- Click on the plus sign next to Monitor in the tree view, then the plus sign next to SLB, then the General link.

The General panel shows the following information.

**Table 6.6: Global Layer 4 Configuration Information**

This Field...	Displays...
Total Connections	The total number of TCP connections the ServerIron is currently managing.
Unsuccessful Connection	The number of client requests for a TCP port that the ServerIron could not fulfill because the server was busy or down, or the port was not configured on the server.
Limit Exceeds	<p>The number of packets dropped by the ServerIron because the TCP SYN limit on the real servers had been reached. The TCP SYN limit is a configurable parameter that allows you to protect servers against TCP SYN attacks by limiting the number of TCP SYN requests the ServerIron can send to the server each second.</p> <p>For more information, see “Modifying Maximum Session Limit” on page 12-58.</p>
Free Session	The number of sessions that are still available for use. By default, the ServerIron is configured to allow the maximum number of sessions it can support. However, if you need to decrease the number of sessions supported, see “Modifying Maximum Session Limit” on page 12-58.
Client->Server	The number of connections initiated by clients.

**Table 6.6: Global Layer 4 Configuration Information (Continued)**

This Field...	Displays...
Server->Client	The number of connections initiated by servers. Generally, this value is 0 unless the client is using FTP or another application that causes the server to initiate connections.
Backup Port	<p>The port number the ServerIron is using for hot standby. If you have not configured hot standby, this field contains the value "None".</p> <p><b>Note:</b> Hot standby and Symmetric SLB are different backup features. You cannot configure both of them on the same ServerIron.</p>

## Displaying Real Server Information

You can display configuration information and statistics for the real servers configured on the ServerIron using either of the following methods.

**USING THE CLI**

To display real server configuration information and statistics, enter the following command at any level of the CLI:

```
ServerIron(config)# show server real
Real Servers Info

Server State - 1:enabled, 2:failed, 3:test, 4:suspect, 5:grace_dn, 6:active
Name:rs1                IP: 209.157.23.60:4    State:6    Wt:1      Max-conn:1000000

Src-nat (cfg:op) = 0: 0 Dest-nat-(cfg:op) = 0: 0
Remote server: No      Dynamic: No
Port  State  Ms  CurConn  TotConns  Rx-pkts  Tx-pkts  Rx-octet  Tx-octet  Reas
pop2  enabled  0      0        0         0        0         0         0        0  0
    Keepalive: Disabled
radiusenabled 0      0        0         0        0         0         0        0  0
    Keepalive: Disabled, Username : "reza"
    Password : "QA", Key : "arvind"
imap4  enabled  0      0        0         0        0         0         0        0  0
    Keepalive: Disabled
ldap   enabled  0      0        0         0        0         0         0        0  0
    Keepalive: Disabled, LDAP Version : 3
70     enabled  6      0        0         0        0         0         0        0  0
    Keepalive: Enabled
dns    enabled  0      0        0         0        0         0         0        0  0
    Keepalive: Disabled, Zone : "foundrynet.com", Addr Query : ""
snmp   enabled  0      0        0         0        0         0         0        0  0
    Keepalive: Disabled
http   enabled  6      0        0         0        0         0         0        0  0
    Keepalive: Disabled, status code(s) default (200-299, 401)
    HTTP URL: "HEAD /"
600    unbnd    6      0        0         0        0         0         0        0  0
    Keepalive: Disabled
500    enabled  6      0        0         0        0         0         0        0  0
    Keepalive: Disabled
defaulunbnd 0      0        0         0        0         0         0         0        0  0

Server Total          0        0         0        0         0         0         0        0  0

information for remaining real servers omitted for brevity..
```

This display shows the following information.

**Table 6.7: Real Server Information**

This Field...	Displays...
<b>Server State Codes</b>	
Server State	The possible values for the server state. The state of each real server is shown by the State field. See below.
<b>General Server Parameters</b>	
Name	The name of the real server. This is the name you assigned to the server when you configured it on the ServerIron.

**Table 6.7: Real Server Information (Continued)**

This Field...	Displays...
IP	<p>The IP address of the real server.</p> <p>If you configured a host range of VIPs on the server, the number following the IP address (after the colon) is the number of hosts on the server. In the example shown above, the VIP address is 209.157.23.60 and the address has been configured with a host range of four hosts. For more information, see “Web Hosting with Unlimited Virtual IP Addresses” on page 6-101.</p>
State	<p>The state of the real server. The state can be one of the following states, also listed next to “Server State” at the top of the <b>show server real</b> display:</p> <ul style="list-style-type: none"> <li>• 1 – Enabled</li> <li>• 2 – Failed</li> <li>• 3 – Test</li> <li>• 4 – Suspect</li> <li>• 5 – Graceful shutdown</li> <li>• 6 – Active</li> </ul>
Wt	<p>The weight assigned to this server. The weight applies only if the predictor (load balancing method) is “weighted”. See “Load Balancing Method (Predictor)” on page 6-55.</p>
Max-conn	<p>The maximum number of client connections that the ServerIron will manage for the server. A connection consists of two sessions, the client-to-server session and the server-to-client session.</p> <p>By default, the ServerIron allows up to 500,000 connections (one million sessions) on a server.</p> <p>If you need to lower the maximum number of connections the ServerIron will manage, see “Modifying Maximum Session Limit” on page 12-58.</p>
Src-nat	<p>The configured and operational states of the source NAT feature. The two states are separated by a colon ( : ). The configured state is shown first, followed by the operational state. Each state can have one of the following values:</p> <ul style="list-style-type: none"> <li>• 0 – Disabled</li> <li>• 1 – Enabled</li> </ul>
Dest-nat	<p>The configured and operational states of the destination NAT feature. The two states are separated by a colon ( : ). The configured state is shown first, followed by the operational state. Each state can have one of the following values:</p> <ul style="list-style-type: none"> <li>• 0 – Disabled</li> <li>• 1 – Enabled</li> </ul>



Table 6.7: Real Server Information (Continued)

This Field...	Displays...
Remote server	<p>Indicates whether the server is configured on the ServerIron as a remote server or a local server. The ServerIron uses remote servers as failovers if all the local servers are down. This field can have one of the following values:</p> <ul style="list-style-type: none"> <li>No – The server is not a remote server.</li> <li>Yes – The server is a remote server.</li> </ul> <p>For more information about remote servers, see “Web Hosting with Geographically-Distributed Servers” on page 6-110.</p>
Dynamic	A statistic used by Foundry technical support.

**TCP/UDP Port Statistics**

The following fields apply to all the TCP/UDP ports on the real servers.

**Note:** For DNS, HTTP, and RADIUS ports, the server-specific health check information for the port is listed under the port’s statistics. For information about the health check parameters, see “Modifying the HTTP Keepalive Method, Value, and Status Codes” on page 12-32.

Port	<p>The TCP/UDP port name or number. This field can have one of the following values:</p> <ul style="list-style-type: none"> <li>default</li> <li>dns – the well-known name for port 53</li> <li>ftp – the well-known name for port 21. (ports 20 and 21 both are FTP ports but on the ServerIron, the name “ftp” corresponds to port 21.)</li> <li>http – the well-known name for port 80</li> <li>imap4 – the well-known name for port 143</li> <li>ldap – the well-known name for port 389</li> <li>nntp – the well-known name for port 119</li> <li>ntp – the well-known name for port 123</li> <li>pop2 – the well-known name for port 109</li> <li>pop3 – the well-known name for port 110</li> <li>radius – the well-known name for udp port 1812</li> <li>smtp – the well-known name for port 25</li> <li>snmp – the well-known name for port 161</li> <li>ssl – the well-known name for port 443</li> <li>telnet – the well-known name for port 23</li> <li>tftp – the well-known name for port 69</li> <li>&lt;number&gt; – the port number, if the port is not one of those listed above</li> </ul>
------	---

Table 6.7: Real Server Information (Continued)

This Field...	Displays...
State	<p>The state of the port. The state can be one of the following:</p> <ul style="list-style-type: none"> <li>• enabled</li> <li>• failed</li> <li>• test</li> <li>• suspect</li> <li>• graceful shutdown</li> <li>• active</li> <li>• unbnd</li> </ul> <p><b>Note:</b> If the state is unbnd, you have not bound the port to a virtual server port.</p>
Ms	<p>The master port state. This field applies only to track ports and to ports to which you have bound other TCP/UDP ports in many-to-one configurations.</p> <ul style="list-style-type: none"> <li>• For track ports, the state of the master port. When a port is configured to track a master port, the ServerIron sends a client's request for the tracking port to the same real server as the master port. See "Track Port Group" on page 6-57 and "TCP/UDP Application Groups" on page 6-104. In the example <b>show real server</b> output shown above, assume that port 500 is tracked by port 600. If port 500's state changes, port 600's state also changes to match.</li> <li>• For many-to-one TCP/UDP port binding, the state of the port that is translated in the port binding between the real server and the virtual server. The ports that are not translated follow the state of the port that is translated. See "Many-To-One TCP/UDP Port Binding" on page 6-98. In the example <b>show real server</b> output shown above, assume that port 70 is untranslated and follows the state of port http. If port http's state changes, port 70's state also changes to match.</li> </ul> <p>This field can have one of the following values for the types of ports listed above:</p> <ul style="list-style-type: none"> <li>• 1 – Enabled</li> <li>• 2 – Failed</li> <li>• 3 – Test</li> <li>• 4 – Suspect</li> <li>• 5 – Graceful shutdown</li> <li>• 6 – Active</li> </ul> <p>For all other types of ports, the value is always 0.</p>
CurConn	<p>The number of client connections currently on the server. A connection consists of two sessions, the client-to-server session and the server-to-client session.</p>

**Table 6.7: Real Server Information (Continued)**

This Field...	Displays...
TotConns	The number of client connections on the server since the ServerIron was last booted. A connection consists of two sessions, the client-to-server session and the server-to-client session.
Rx-pkts	The number of packets the ServerIron has received from the server.
Tx-pkts	The number of packets the ServerIron has sent to the server.
Rx-octet	The number of octets (bytes) the ServerIron has received from the server.
Tx-octet	The number of octets (bytes) the ServerIron has sent to the server.
Reas	<p>The number of times the ServerIron has reassigned the connection to another server in the rotation because the server that is in use has not responded to two contiguous TCP SYNs from the client. When this occurs, the ServerIron directs the client to another server upon receiving the third SYN from the client.</p> <p><b>Note:</b> Windows 98 sends two TCP SYNs for each connection attempt.</p> <p><b>Note:</b> This statistic does not apply to SwitchBack (Direct Server Return).</p>

**USING THE WEB MANAGEMENT INTERFACE**

To display information for a real server configured on the ServerIron, do one of the following:

- Select the Real Server link from the list of statistics links at the bottom of the General panel or other SLB panels that have the links.
- Click on the plus sign next to Monitor in the tree view, then the plus sign next to SLB, then the Real Server link.

**NOTE:** If data entry fields for a real server are displayed instead of a table listing real servers, then there are no real servers configured on the ServerIron.

The Real Server panel shows the following information.

**Table 6.8: Real Server Information**

This Field...	Displays...
Server Name	The name of the real server. This is the name you assigned to the server when you configured it on the ServerIron.
Server IP	The IP address of the real server.

**Table 6.8: Real Server Information (Continued)**

This Field...	Displays...
State	<p>The state of the real server. The state can be one of the following:</p> <ul style="list-style-type: none"> <li>• active</li> <li>• enabled</li> <li>• failed</li> <li>• grace_dn</li> <li>• suspect</li> <li>• test</li> </ul>
Current Connection	The number of client connections currently on the server. A connection consists of two sessions, the client-to-server session and the server-to-client session.
Peak Connection	The highest number of simultaneous client connections on the server since the ServerIron was booted. A connection consists of two sessions, the client-to-server session and the server-to-client session.
Total Connection	The number of client connections on the server since the ServerIron was last booted. A connection consists of two sessions, the client-to-server session and the server-to-client session.
Rx Pkts	The number of packets the ServerIron has received from the server.
Tx Pkts	The number of packets the ServerIron has sent to the server.
Rx Bytes	The number of bytes (octets) the ServerIron has received from the server.
Tx Bytes	The number of bytes (octets) the ServerIron has sent to the server.
Rx Avg Frame	The average frame size of the packets received by the ServerIron from the server.
Tx Avg Frame	The average frame size of the packets sent to the server.
Age	<p>The total number of TCP and UDP sessions that the ServerIron closed because the aged out. A session ages out when the age timer configured on the ServerIron expires. For more information, see “Modifying the TCP Age” on page 12-59 and “Modifying the UDP Age” on page 12-59.</p>
Reassignments	<p>The number of times the ServerIron has reassigned the connection to another server in the rotation because the server that is in use has not responded to two TCP SYNs from the client.</p> <p><b>Note:</b> Windows 98 sends two TCP SYNs for each connection attempt.</p>
Failed Port Exist	The number of times a client request could not be fulfilled because the client requested a port that is not configured on the server.
Fail Time	A statistic used by Foundry technical support.

To display information for a real server port, do one of the following:

- Select the [Real Server Port](#) link from the list of statistics links at the bottom of the General panel or other SLB panels that have the links.
- Click on the plus sign next to Monitor in the tree view, then the plus sign next to SLB, then the [Real Server Port](#) link.

**NOTE:** If data entry fields for a real server port are displayed instead of a table listing real server ports, then there are no real server ports configured on the ServerIron.

The Real Server Port panel shows the following information.

**Table 6.9: Real Server Port Information**

This Field...	Displays...
Server Name	The name of the real server. This is the name you assigned to the server when you configured it on the ServerIron.
TCP/UDP Port	<p>The TCP/UDP port name or number. This field can have one of the following values:</p> <ul style="list-style-type: none"> <li>• default</li> <li>• dns – the well-known name for port 53</li> <li>• ftp – the well-known name for port 21. (ports 20 and 21 both are FTP ports but on the ServerIron, the name “ftp” corresponds to port 21.)</li> <li>• http – the well-known name for port 80</li> <li>• imap4 – the well-known name for port 143</li> <li>• ldap – the well-known name for port 389</li> <li>• nntp – the well-known name for port 119</li> <li>• ntp – the well-known name for port 123</li> <li>• pop2 – the well-known name for port 109</li> <li>• pop3 – the well-known name for port 110</li> <li>• radius – the well-known name for udp port 1812</li> <li>• smtp – the well-known name for port 25</li> <li>• snmp – the well-known name for port 161</li> <li>• ssl – the well-known name for port 443</li> <li>• telnet – the well-known name for port 23</li> <li>• tftp – the well-known name for port 69</li> <li>• &lt;number&gt; – the port number, if the port is not one of those listed above</li> </ul>
Current Connection	The number of client connections currently on the server. A connection consists of two sessions, the client-to-server session and the server-to-client session.

**Table 6.9: Real Server Port Information (Continued)**

This Field...	Displays...
Peak Connection	The highest number of simultaneous client connections on the server since the ServerIron was booted. A connection consists of two sessions, the client-to-server session and the server-to-client session.
Total Connection	The number of client connections on the server since the ServerIron was last booted. A connection consists of two sessions, the client-to-server session and the server-to-client session.
Rx Pkts	The number of packets the ServerIron has received from the server.
Tx Pkts	The number of packets the ServerIron has sent to the server.
Rx Bytes	The number of bytes (octets) the ServerIron has received from the server.
Tx Bytes	The number of bytes (octets) the ServerIron has sent to the server.
Rx Avg Frame	The average frame size of the packets received by the ServerIron from the server.
Tx Avg Frame	The average frame size of the packets sent to the server.
Age	The total number of TCP and UDP sessions that the ServerIron closed because the aged out. A session ages out when the age timer configured on the ServerIron expires. For more information, see "Modifying the TCP Age" on page 12-59 and "Modifying the UDP Age" on page 12-59.
Reassignments	<p>The number of times the ServerIron has reassigned the connection to another server in the rotation because the server that is in use has not responded to two TCP SYNs from the client.</p> <p><b>Note:</b> Windows 98 sends two TCP SYNs for each connection attempt.</p>
State	<p>The state of the port. The state can be one of the following:</p> <ul style="list-style-type: none"> <li>• enabled</li> <li>• failed</li> <li>• test</li> <li>• suspect</li> <li>• graceful shutdown</li> <li>• active</li> <li>• unbnd</li> </ul> <p><b>Note:</b> If the state is unbnd, you have not bound the port to a virtual server port.</p>
Fail Time	A statistic used by Foundry technical support.

## Displaying Virtual Server Information

You can display configuration information and statistics for the virtual servers configured on the ServerIron using either of the following methods.

### USING THE CLI

To display virtual server configuration information and statistics, enter the following command at any level of the CLI:

```
ServerIron(config)# show server virtual
Virtual Servers Info

Server Name: v100                IP : 209.157.23.100 : 4
Status: enabled Predictor: least-conn TotConn: 4233
Dynamic: No HTTP redirect: disabled
Sym: group = 1 state = 5 priority = 2 keep = 0
  Activates = 4, Inactive= 3
Port    State    Sticky  Concur    CurConn    TotConn    PeakConn
radius-oenabled NO      NO      0          0          0
http    enabled NO      NO      0          4233       39
ftp     enabled NO      NO      0          0          0
telnet  enabled NO      NO      0          0          0
ssl     enabled YES    NO      0          0          0
smtp    enabled NO      NO      0          0          0
nntp    enabled NO      NO      0          0          0
ntp     enabled NO      NO      0          0          0
dns     enabled NO      NO      0          0          0
pop2    enabled NO      NO      0          0          0
pop3    enabled NO      NO      0          0          0
tftp    enabled NO      NO      0          0          0
imap4   enabled NO      NO      0          0          0
snmp    enabled NO      NO      0          0          0
ldap    enabled NO      NO      0          0          0
default enabled NO      NO      0          0          0
```

*information for remaining virtual servers omitted for brevity...*

This display shows the following information.

**Table 6.10: Virtual Server Information**

This Field...	Displays...
<b>General Server Parameters</b>	
Server Name	The name of the virtual server. This is the name you assigned to the server when you configured it on the ServerIron.
IP	The IP address of the virtual server.  If you configured a host range of VIPs on the server, the number following the IP address (after the colon) is the number of hosts on the server. In the example above, the VIP has a host range of 4 addresses.
Status	The status of the virtual server. The status can be one of the following: <ul style="list-style-type: none"> <li>enabled</li> <li>disabled</li> </ul>

**Table 6.10: Virtual Server Information (Continued)**

This Field...	Displays...
Predictor	<p>The load balancing predictor the ServerIron uses to balance traffic among the real servers bound to this virtual server. The predictor can be one of the following:</p> <ul style="list-style-type: none"> <li>least-conn (least connections)</li> <li>least-sess (least sessions)</li> <li>response-time (server response time)</li> </ul> <p><b>Note:</b> This value also applies to the combined method of least connections and server response time weights.</p> <ul style="list-style-type: none"> <li>round-robin (round robin)</li> <li>weighted (weighted percentage)</li> <li>least-local-conn (least local connections)</li> <li>least-local-sess (least local sessions)</li> </ul> <p>You can assign these metrics on a global basis and an individual virtual server basis.</p> <p>For more information or to globally change the predictor, see “Load Balancing Method (Predictor)” on page 6-24.</p> <p>To locally change the predictor for a virtual server, see “Load Balancing Method (Predictor)” on page 6-55.</p>
TotConn	<p>The number of client connections on the server since the ServerIron was last booted or restarted. A connection consists of two sessions, the client-to-server session and the server-to-client session.</p>
Dynamic	<p>A statistic used by Foundry technical support.</p>
HTTP-redirect	<p>The state of the HTTP redirect feature. This feature enables the ServerIron to send an HTTP redirect message to the client if all the real servers are down and the ServerIron is therefore sending client requests to a remote server.</p> <p>The HTTP redirect message instructs the client to redirect its HTTP connection directly to the remote server, bypassing the ServerIron.</p> <p>The state can be one of the following:</p> <ul style="list-style-type: none"> <li>disabled</li> <li>enabled</li> </ul> <p>For more information, see “Using HTTP Redirect with Geographically-Distributed Servers” on page 6-113.</p>



Table 6.10: Virtual Server Information (Continued)

This Field...	Displays...
Sym	<p>Information for Symmetric SLB. The following information is displayed:</p> <ul style="list-style-type: none"> <li>group – The Symmetric SLB group number.</li> <li>state – The state, which should be 5 for the active ServerIron and 3 for other ServerIrons.</li> <li>priority – The Symmetric SLB priority configured on the ServerIron.</li> <li>keep – The number of times an SSLB backup has failed to communicate with the active ServerIron. By default, the counter is incremented by 1 every 400 milliseconds the backup ServerIron is late responding to the active ServerIron's keepalive message. The counter is reset to 0 each time the backup ServerIron replies to a keepalive message. If the counter goes higher than the maximum number allowed (20 by default, thus 8 seconds), the standby ServerIron takes over as the new active ServerIron. Normally, this field almost always contains 0.</li> </ul> <p><b>Note:</b> This field is applicable only on the active ServerIron.</p> <ul style="list-style-type: none"> <li>dyn priority/factor – The current dynamically set priority and the decrement value. In this example, an application has failed a health check, so the dynamic priority is 20 instead of 30. The decrement value is 10. If the priority and dyn priority values match, then all the VIP's applications that are configured for SSLB are responding to their health checks.</li> <li>Activates – The number of times this ServerIron has become the active ServerIron.</li> <li>Inactive – The number of times this ServerIron has changed from being the active ServerIron.</li> <li>Best-standby-mac – The MAC address of the backup ServerIron with the second-highest priority. This ServerIron will become the active ServerIron if a failover occurs.</li> </ul> <p>For more information about Symmetric SLB, see "Configuring Symmetric SLB and SwitchBack" on page 7-1.</p>
<b>TCP/UDP Port Information and Statistics</b>	

**Table 6.10: Virtual Server Information (Continued)**

This Field...	Displays...
Port	<p>The TCP/UDP port name or number. This field can have one of the following values:</p> <ul style="list-style-type: none"> <li>• default</li> <li>• dns – the well-known name for port 53</li> <li>• ftp – the well-known name for port 21. (ports 20 and 21 both are FTP ports but on the ServerIron, the name “ftp” corresponds to port 21.)</li> <li>• http – the well-known name for port 80</li> <li>• imap4 – the well-known name for port 143</li> <li>• ldap – the well-known name for port 389</li> <li>• nntp – the well-known name for port 119</li> <li>• ntp – the well-known name for port 123</li> <li>• pop2 – the well-known name for port 109</li> <li>• pop3 – the well-known name for port 110</li> <li>• radius – the well-known name for udp port 1812</li> <li>• radiuso – UDP port 1645, which is used in some older RADIUS implementations instead of port 1812</li> <li>• smtp – the well-known name for port 25</li> <li>• snmp – the well-known name for port 161</li> <li>• ssl – the well-known name for port 443</li> <li>• telnet – the well-known name for port 23</li> <li>• tftp – the well-known name for port 69</li> <li>• &lt;number&gt; – the port number, if the port is not one of those listed above</li> </ul>
State	<p>The state of the port. The state can be one of the following:</p> <ul style="list-style-type: none"> <li>• enabled</li> <li>• failed</li> <li>• test</li> <li>• suspect</li> <li>• graceful shutdown</li> <li>• active</li> <li>• unbnd</li> </ul> <p><b>Note:</b> If the status is unbnd, you have not bound the port to a real server port.</p>

**Table 6.10: Virtual Server Information (Continued)**

This Field...	Displays...
Sticky	<p>Whether the port is “sticky”. When a port is sticky, the ServerIron uses the same real server for multiple requests from the same client for the port. For non-sticky ports, the ServerIron load balances the requests and thus does not necessarily send them all to the same real server.</p> <p>This parameter can have one of the following values:</p> <ul style="list-style-type: none"> <li>• NO</li> <li>• YES</li> </ul> <p>For more information, see “TCP/UDP Application Groups” on page 6-104.</p>
Concur	<p>Whether the port is configured for concurrent connections. A port configured to allow concurrent connections can have more than connection open to the same client at the same time.</p> <p>For more information, see “TCP/UDP Application Groups” on page 6-104.</p>
CurConn	The number of client connections currently on the server. A connection consists of two sessions, the client-to-server session and the server-to-client session.
TotConn	The number of client connections on the server since the ServerIron was booted. A connection consists of two sessions, the client-to-server session and the server-to-client session.
PeakConn	The highest number of connections the VIP has had at the same time.

**USING THE WEB MANAGEMENT INTERFACE**

To display information for a virtual server configured on the ServerIron, do one of the following:

- Select the [Virtual Server](#) link from the list of statistics links at the bottom of the General panel or other SLB panels that have the links.
- Click on the plus sign next to Monitor in the tree view, then the plus sign next to SLB, then the [Virtual Server](#) link.

**NOTE:** If data entry fields for a virtual server are displayed instead of a table listing virtual servers, then there are no virtual servers configured on the ServerIron.

The Virtual Server panel shows the following information.

**Table 6.11: Virtual Server Information**

This Field...	Displays...
Server Name	The name of the virtual server. This is the name you assigned to the server when you configured it on the ServerIron.
Server IP	The IP address of the virtual server.

**Table 6.11: Virtual Server Information (Continued)**

This Field...	Displays...
Metrics	<p>The load balancing metric the ServerIron uses to balance traffic among the real servers bound to this virtual server. The predictor can be one of the following:</p> <ul style="list-style-type: none"> <li>least-conn (least connections)</li> <li>response-time (server response time)</li> </ul> <p><b>Note:</b> This value also applies to the combined method of least connections and server response time weights.</p> <ul style="list-style-type: none"> <li>round-robin (round robin)</li> <li>weighted (weighted percentage)</li> </ul> <p>You can assign these metrics on a global basis and an individual virtual server basis.</p> <p>For more information or to globally change the predictor, see “Load Balancing Method (Predictor)” on page 6-24.</p> <p>To locally change the predictor for a virtual server, see “Load Balancing Method (Predictor)” on page 6-55.</p>
Symmetric	<p>Information for Symmetric SLB. The following information is displayed:</p> <ul style="list-style-type: none"> <li>Group – the Symmetric SLB group number</li> <li>State – the state, which should be 5 for the active ServerIron and 3 for other ServerIrons</li> <li>Priority – the Symmetric SLB priority</li> <li>Keep – a statistic used by Foundry technical support</li> <li>Activates – the number of times this ServerIron has become the active ServerIron</li> <li>Inactive – the number of times this ServerIron has changed from being the active ServerIron</li> </ul> <p>For more information about Symmetric SLB, see “Configuring Symmetric SLB and SwitchBack” on page 7-1.</p>
CurConn	The number of client connections currently on the server. A connection consists of two sessions, the client-to-server session and the server-to-client session.
TotConns	The number of client connections on the server since the ServerIron was last booted. A connection consists of two sessions, the client-to-server session and the server-to-client session.
Rx pkts	The number of packets the ServerIron has received from the server.
Tx Pkts	The number of packets the ServerIron has sent to the server.
Rx Bytes	The number of bytes (octets) the ServerIron has received from the server.
Tx Bytes	The number of bytes (octets) the ServerIron has sent to the server.

**Table 6.11: Virtual Server Information (Continued)**

This Field...	Displays...
Rx Avg Frame	The average frame size of the packets received by the ServerIron from the server.
Tx Avg Frame	The average frame size of the packets sent to the server.

To display information for a virtual server port, do one of the following:

- Select the [Virtual Server Port](#) link from the list of statistics links at the bottom of the General panel or other SLB panels that have the links.
- Select the [Show](#) link to display the Show Statistics panel, then select the [Virtual Server Port](#) link.

**NOTE:** If data entry fields for a virtual server port are displayed instead of a table listing virtual server ports, then there are no virtual server ports configured on the ServerIron.

The Virtual Server Port panel shows the following information.

**Table 6.12: Virtual Server Port Information**

This Field...	Displays...
Server Name	The name of the virtual server. This is the name you assigned to the server when you configured it on the ServerIron.

**Table 6.12: Virtual Server Port Information (Continued)**

This Field...	Displays...
TCP/UDP Port	<p>The TCP/UDP port name or number. This field can have one of the following values:</p> <ul style="list-style-type: none"> <li>• default</li> <li>• dns – the well-known name for port 53</li> <li>• ftp – the well-known name for port 21. (ports 20 and 21 both are FTP ports but on the ServerIron, the name “ftp” corresponds to port 21.)</li> <li>• http – the well-known name for port 80</li> <li>• imap4 – the well-known name for port 143</li> <li>• ldap – the well-known name for port 389</li> <li>• nntp – the well-known name for port 119</li> <li>• ntp – the well-known name for port 123</li> <li>• pop2 – the well-known name for port 109</li> <li>• pop3 – the well-known name for port 110</li> <li>• radius – the well-known name for udp port 1812</li> <li>• smtp – the well-known name for port 25</li> <li>• snmp – the well-known name for port 161</li> <li>• ssl – the well-known name for port 443</li> <li>• telnet – the well-known name for port 23</li> <li>• tftp – the well-known name for port 69</li> <li>• &lt;number&gt; – the port number, if the port is not one of those listed above</li> </ul>
Status	<p>The status of the port. The status can be one of the following:</p> <ul style="list-style-type: none"> <li>• enabled</li> <li>• failed</li> <li>• test</li> <li>• suspect</li> <li>• graceful shutdown</li> <li>• active</li> <li>• unbnd</li> </ul> <p><b>Note:</b> If the status is unbnd, you have not bound the port to a real server port.</p>

Table 6.12: Virtual Server Port Information (Continued)

This Field...	Displays...
Sticky	<p>Whether the port is “sticky”. When a port is sticky, the ServerIron uses the same real server for multiple requests from the same client for the port. For non-sticky ports, the ServerIron load balances the requests and thus does not necessarily send them all to the same real server.</p> <p>This parameter can have one of the following values:</p> <ul style="list-style-type: none"> <li>• No</li> <li>• Yes</li> </ul> <p>For more information, see “TCP/UDP Application Groups” on page 6-104.</p>
Concurrent	<p>Whether the port is configured for concurrent connections. A port configured to allow concurrent connections can have more than connection open to the same client at the same time.</p> <p>For more information, see “TCP/UDP Application Groups” on page 6-104.</p>
Translate	<p>Whether port translation is enabled. This parameter can have one of the following values:</p> <ul style="list-style-type: none"> <li>• No</li> <li>• Yes</li> </ul> <p>The value should always be Yes except for ports configured for many-to-one port binding. See “Many-To-One TCP/UDP Port Binding” on page 6-98.</p>
DSR	<p>Whether SwitchBack is enabled for the port. SwitchBack causes the real server to send responses directly to the client instead of sending them back through the ServerIron. See “Using SwitchBack” on page 7-15.</p>

## Displaying Port-Binding Information

You can display port-binding information using either of the following methods.

### USING THE CLI

To display port-binding information, enter the following command at any level of the CLI:

```
ServerIron(config)# show server bind
Virtual Server Name: v100,   IP: 209.157.23.100
  http -----> s43: 209.157.23.43,  http
                  s60: 209.157.23.60,  8080
  ftp -----> s43: 209.157.23.43,  ftp
                  s60: 209.157.23.60,  ftp
  70 -----> s43: 209.157.23.43,  70
                  s60: 209.157.23.60,  70
Virtual Server Name: v105,   IP: 209.157.23.105
  telnet -----> s60: 209.157.23.60,  300
  ftp -----> s60: 209.157.23.60,  200
  http -----> s60: 209.157.23.60,  100
  dns -----> s60: 209.157.23.60,  400
  tftp -----> s60: 209.157.23.60,  500
```

The display lists the port bindings for each virtual server configured on the ServerIron. The first row of information for each virtual server lists the virtual server name and VIP. The following rows list the TCP/UDP ports configured on the virtual server and the real servers and port names or numbers to which each port is bound.

In the example shown above, two virtual servers are configured on the ServerIron, v100 and v105. The first set of rows in the example output shown above is for virtual server v100, with VIP 209.157.23.100.

The rows below the first row list the real servers and ports to which the virtual server's ports are bound. The rows are grouped by port type. The first two rows after the first row in the example above list the port bindings for the virtual server's HTTP port. In this case, the virtual server is bound to an HTTP port on two real servers, s43 and s60. The port name or number on the real server is listed following the real server's IP address. In this example, the HTTP port to which v100 is bound on s43 is "http", the well-known name for the port. The virtual server's HTTP port is bound to port 8080 on real server s60.

### USING THE WEB MANAGEMENT INTERFACE

To display binding information, do one of the following:

- Select the [Bind](#) link from the list of links at the bottom of the General panel or another SLB panel that has the link.
- Click on the plus sign next to Configure in the tree view, then the plus sign next to SLB, then the [Bind](#) link.

The Bind panel shows the following information.

**Table 6.13: Port Binding Information**

This Field...	Displays...
Virtual Server Name	The name of the virtual server. This is the name you assigned to the server when you configured it on the ServerIron.
Virtual TCP/UDP Port	The TCP/UDP port name or number on the virtual server.
Real Server Name	The name of the real server. This is the name you assigned to the server when you configured it on the ServerIron.
Real TCP/UDP Port	The TCP/UDP port name or number on the real server.



## Displaying Session Statistics

You can display global and real-server session statistics using either of the following methods.

### USING THE CLI

To display port-binding information, enter the following command at any level of the CLI:

```
ServerIron(config)# show server sessions
Avail. Sessions      =      524287  Total Sessions      =      524288
Total C->S Conn      =      4233   Total S->C Conn      =          0
Total Reassign       =          0  Unsuccessful Conn    =          0
Server State - 1:enabled, 2:failed, 3:test, 4:suspect, 5:grace_dn, 6:active

Real Server    State  CurrConn  TotConn TotRevConn  CurrSess  PeakConn
s60            1      0          0          0          0          0
s43            1      0      4233          0          0          39
```

This display shows the following information.

**Table 6.14: Session Statistics**

This Field...	Displays...
<b>Global Statistics</b>	
Avail. Sessions	The number of sessions that are still available for use. By default, the ServerIron is configured to allow the maximum number of sessions it can support. However, if you need to decrease the number of sessions supported, see “Modifying Maximum Session Limit” on page 12-58.
Total Sessions	The number of sessions that are currently in use.
Total C->S Conn	The number of connections initiated by clients.
Total S->C Conn	The number of connections initiated by servers. Generally, this value is 0 unless the client is using FTP or another application that causes the server to initiate connections.
Total Reassign	<p>The number of unacknowledged TCP SYN-ACKs on all the real servers combined. When a server reaches the maximum number of unacknowledged TCP SYN-ACKs allowed by the ServerIron (the reassign threshold), the ServerIron marks that server FAILED and removes it from the load balancing rotation.</p> <p>The TCP SYN-ACK counter increments only when acknowledgments are not received. Each time an expected TCP SYN-ACK is received from a real server, the counter is cleared for that server, thus reducing the total count.</p> <p>For more information, see “Modifying the Reassign Threshold” on page 12-29.</p> <p><b>Note:</b> This statistic does not apply to SwitchBack (Direct Server Return).</p>
Unsuccessful Conn	The number of connection attempts by clients or servers that were unsuccessful.

**Table 6.14: Session Statistics (Continued)**

This Field...	Displays...
<b>Statistics for Individual Real Servers</b>	
Server State	The possible values for the server state. The state of each real server is shown by the State field. See below.
Real Server	The name of the real server. This is the name you gave the server when you configured it.
State	The state of the real server. The state can be one of the states listed by "Server State" at the top of the display.
CurConn	The number of client connections currently on the server. A connection consists of two sessions, the client-to-server session and the server-to-client session.
TotConn	The number of client connections on the server since the ServerIron was last booted or restarted. A connection consists of two sessions, the client-to-server session and the server-to-client session.
Tot RevConn	The total number of connections initiated by the server to a client.
CurrSess	The number of sessions currently open on the ServerIron.
PeakConn	The highest number of simultaneous connections the ServerIron has managed since it was last booted or restarted.

#### [USING THE WEB MANAGEMENT INTERFACE](#)

Session statistics are on the global Server Load Balancing statistics panel. See "Displaying Global Configuration Information" on page 6-70.

### Displaying Traffic Statistics

You can display packet statistics for ServerIron traffic using either of the following methods.

#### [USING THE CLI](#)

To display traffic statistics, enter the following command at any level of the CLI:

```
ServerIron(config)# show server traffic
Client->Server      =      26753  Server->Client      =      24817
Drops               =           4  Aged               =           38
Fw_drops            =           0  Rev_drops          =           0
FIN_or_RST          =      8429  old-conn           =           0
Disable_drop        =           0  Exceed_drop        =           0
Stale_drop          =          14  Unsuccessful       =           0
```

This display shows the following information.

**Table 6.15: Traffic Statistics**

This Field...	Displays...
Client->Server	The total number of packets sent from clients to servers.

**Table 6.15: Traffic Statistics (Continued)**

This Field...	Displays...
Server->Client	The total number of packets sent from servers to clients.
Drops	<p>The total number of packets dropped by the ServerIron. This statistic includes the following:</p> <ul style="list-style-type: none"> <li>• TCP Resets – Resets sent by the ServerIron</li> <li>• Forward Resets – Resets from the client</li> <li>• Unsuccessful requests – Requests sent to a TCP or UDP port that is not bound to the request's destination VIP</li> </ul>
Aged	The total number of TCP and UDP sessions that the ServerIron closed because they aged out. A session ages out when the age timer configured on the ServerIron expires. For more information, see "Modifying the TCP Age" on page 12-59 and "Modifying the UDP Age" on page 12-59.
Fw_drops	<p>The number of client-to-server packets the ServerIron has dropped. If this statistic is high, there might not be a session entry. This can occur under the following circumstances:</p> <ul style="list-style-type: none"> <li>• If the session is terminated normally but the client sends another RESET.</li> <li>• If Denial of Service (DoS) protection is configured and has been activated.</li> <li>• If the maximum number of sessions has been reached.</li> <li>• If all the real servers are down.</li> </ul>
Rev_drops	The number of server-to-client packets the ServerIron has dropped. If this statistic is high, there might not be a session entry. This can occur for the same reasons as listed for the Fw_drops field.
FIN_or_RST	A statistic used by Foundry technical support.
old-conn	A statistic used by Foundry technical support.
Disable_drop	The number of packets the ServerIron dropped because they were sent by a client to a VIP port that is bound to a real server port that is currently disabled.
Exceed_drop	<p>The number of packets dropped by the ServerIron because the TCP SYN limit on the real servers had been reached. The TCP SYN limit is a configurable parameter that allows you to protect servers against TCP SYN attacks by limiting the number of TCP SYN requests the ServerIron can send to the server each second.</p> <p>For more information, see "Modifying Maximum Session Limit" on page 12-58.</p>
Stale_drop	The number of TCP SYN packets the ServerIron dropped because they matched a stale session entry.

**Table 6.15: Traffic Statistics (Continued)**

This Field...	Displays...
Unsuccessful	<p>The number of packets that were dropped for one of the following reasons:</p> <ul style="list-style-type: none"> <li>A deny filter configured on the ServerIron matched the packet, causing the ServerIron to drop the packet.</li> <li>A client requested a TCP/UDP port that is not bound on the VIP.</li> </ul>

#### USING THE WEB MANAGEMENT INTERFACE

Traffic statistics are on the global Server Load Balancing statistics panel. See “Displaying Global Configuration Information” on page 6-70.

## SLB Application Examples

The examples in this section illustrate implementations of the following features.

**NOTE:** For configuration examples for Symmetric SLB and SwitchBack, see “Configuring Symmetric SLB and SwitchBack” on page 7-1.

**Table 6.16: SLB Application Examples**

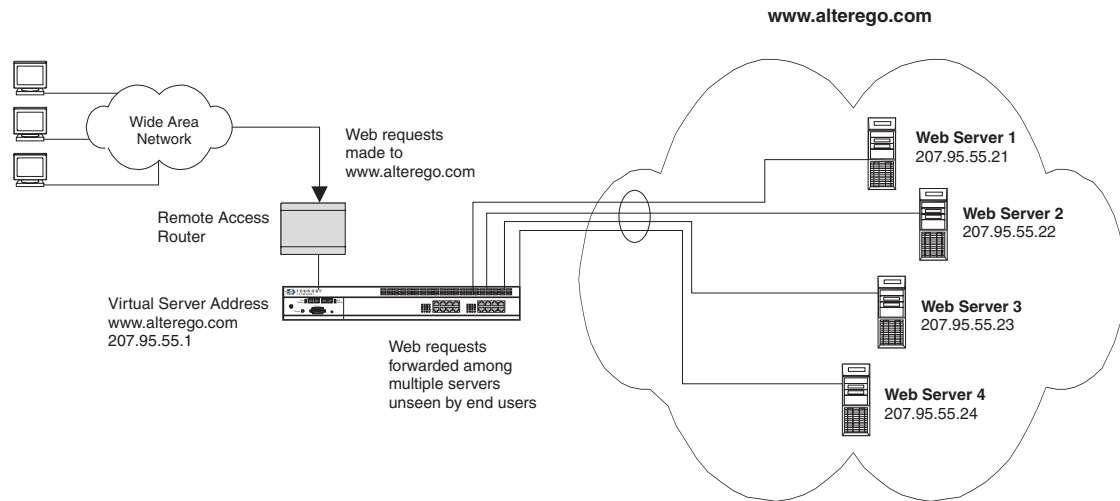
Application	See page...
Basic SLB configuration, mapping one virtual server to multiple real servers	6-97
Mapping multiple virtual servers to one real server (with separate TCP/UDP port binding for each VIP)	6-97
Many-to-one TCP/UDP port bindings	6-98
Mapping a range of contiguous VIPs to contiguous ranges on real servers (unlimited VIP feature)	6-101
Mapping multiple VIPs to multiple real servers for multiple TCP/UDP applications	6-104
Grouping TCP/UDP ports together to keep applications for a client on the same real server (configurable application groups)	6-104
Deploying the ServerIron in a multinetted environment	6-107
Using remote servers as failovers if local servers are unavailable (geographically-distributed servers)	6-110
Using HTTP redirect to send client connections directly to remote real servers	6-113
Using the Reverse Proxy SLB feature to send web requests to a cache server, then to a load balanced web server if the cache does not have the requested content	6-114
Load balancing requests for streaming media files	6-119

**NOTE:** For configuration information and examples of SwitchBack, see “Configuring Symmetric SLB and SwitchBack” on page 7-1.

## Web Hosting with One Virtual Server Mapped to Multiple Real Servers

Suppose a company establishes a web site with a URL of [www.alterego.com](http://www.alterego.com). The company system administrator configures the networks so that the SLB switch forwards web requests among four independent servers, as shown in Figure 6.14.

**Figure 6.14** Real and virtual server assignments in a backbone ServerIron network



Domain Name	Virtual IP	TCP Port	Real IP	TCP Port
www.alterego.com	207.95.55.1	80	207.95.55.21 (web1)	80
			207.95.55.22 (web2)	80
			207.95.55.23 (web3)	80
			207.95.55.24 (web4)	80

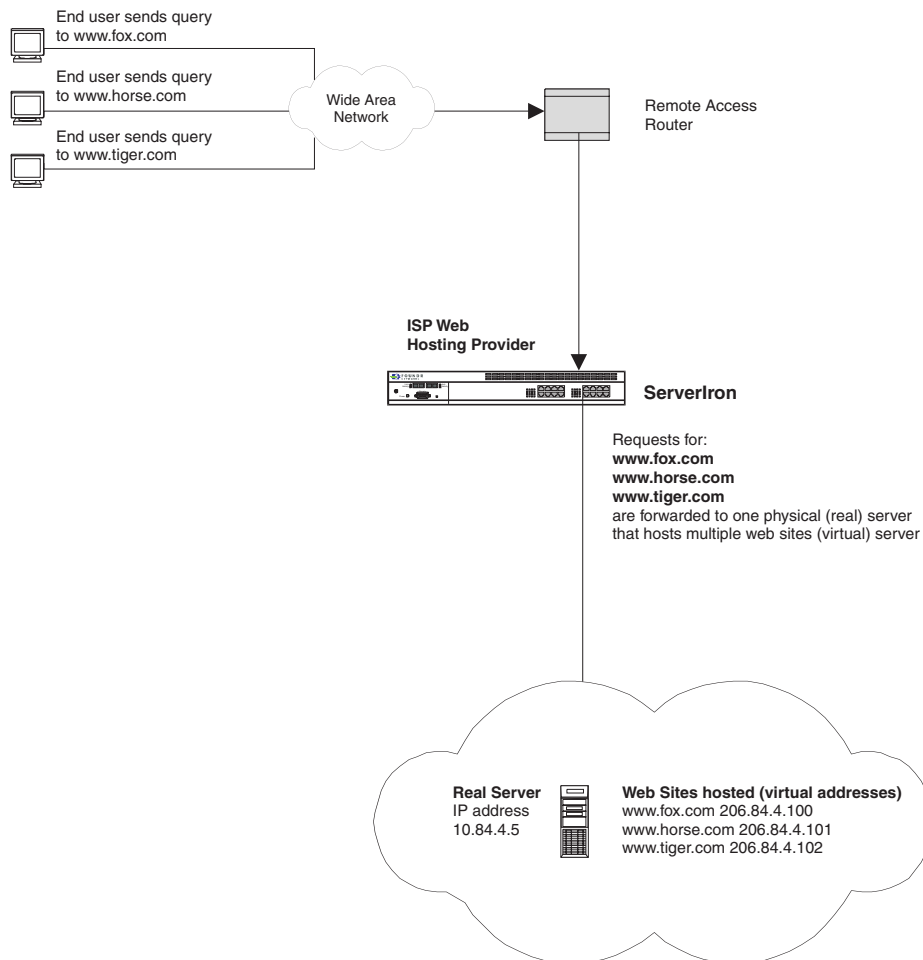
## Web Hosting with Multiple Virtual Servers Mapped to One Real Server

Suppose an ISP administrator wants to use one real server to accommodate three premium users, all of which are web sites. Each of these premium users is assigned its own web site URL:

- [www.fox.com](http://www.fox.com)
- [www.horse.com](http://www.horse.com)
- [www.tiger.com](http://www.tiger.com)

As shown in Figure 6.15, the SLB switch forwards requests received for each of the three web sites to the real server(s) assigned to handle the traffic.

**Figure 6.15 One real server hosting multiple virtual servers**



## Many-To-One TCP/UDP Port Binding

Most SLB configurations for web hosting map one virtual IP address to multiple real servers. However, suppose an ISP wants to host one or multiple domain names on the same real server, using the same TCP/UDP port *and* use a different VIP for each site. Using a separate VIP for each web site eases administration and accounting by allowing the ISP to display statistics on the ServerIron for each VIP address. In addition, you can create the appearance that you have many DNS servers even if you have only a few.

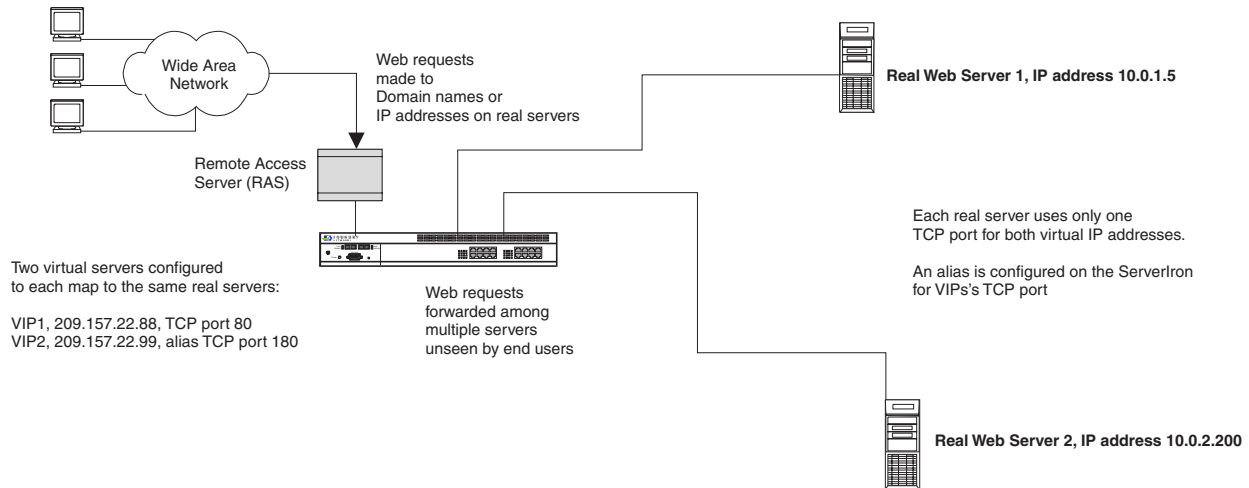
When you bind a port on a real server with a port on a virtual server together, the ServerIron makes an entry in its internal Layer 4 binding table. The port on the real server cannot be bound again to another virtual server if the Layer 4 binding table already has a binding for that port. Thus, to map multiple VIPs to the same real server, normally you need to map each VIP to a different TCP/UDP port on the real server.

If you want to bind multiple VIPs to the same TCP/UDP port on the same real server for accounting reasons, you can do so by creating an alias for the port. When you create an alias, you configure the VIP to bind to a different port number on the real server, then disable port translation for that binding. The ServerIron thus collects and presents information for the alias port number, but traffic from all the VIPs goes to the same TCP/UDP port number on the real server.

To map multiple virtual IP addresses to the same real server, disable HTTP port translation for all but one of the virtual IP addresses, then bind the virtual IP addresses to an alias HTTP port. Disabling HTTP port translation enables the virtual IP addresses to use the same actual HTTP port number on the real server while the ServerIron collects and displays separate statistics for the alias HTTP port number associated with each virtual IP address.

Figure 6.16 shows an example of this type of configuration.

**Figure 6.16 Multiple virtual IP addresses mapped to the same real server**



Virtual Domain Name	Virtual IP	TCP Port	Real IP	TCP Port
www.travel.com	209.157.22.88	80	S1: 10.0.1.5 S2: 10.0.2.200	80
www.weather.com	209.157.22.99	80	S1: 10.0.1.5 S2: 10.0.2.200	180

### Configuration Rules

Use the following rules when configuring the ServerIron to bind more than one virtual server to the same real server using the same application port:

- You must configure both the real port and the alias port on the real server(s). For example, if you need to create alias port 180 so that you can bind two virtual servers to the same real server and application port (80) on a real server, you must configure port 80 and port 180 on the real server. Otherwise, you will not be able to completely bind all the virtual servers to the real server. In the example below, the following real server configurations are incomplete because neither of the real servers has both the untranslated and alias ports configured:

```
ServerIron(config)# server real-name r1 10.0.1.5
ServerIron(config-rs-r1)# port http
ServerIron(config-rs-r1)# exit
ServerIron(config)# server real-name r2 10.0.2.200
ServerIron(config-rs-r2)# port 180
ServerIron(config-rs-r2)# exit
```

- You cannot bind to both the untranslated port and the alias port in the same bind statement. In the example below, the following bind statement is invalid:

```
ServerIron(config-vs-VIP1)# port http
ServerIron(config-vs-VIP1)# bind http r1 http r2 180
```

Here is a more detailed explanation.

When you configure SLB, one of the tasks you perform is to bind the TCP or UDP application ports on the virtual server to their counterparts on the real server. For example, if clients will be sending requests to port 80 (HTTP)

on virtual server `www.mysite.com`, but your real servers expect the HTTP connections to arrive on port 8080 of real server R1, then you must bind port 80 on the virtual server to port 8080 on the real server.

One of the requirements is that a real server can be bound to only one virtual server using the same TCP or UDP application port. Thus, once you bind a real server port to a virtual server port, you cannot bind the same real server port to a different virtual server port.

Normally, the ServerIron translates the IP address and application port of the virtual server requested by the client into the real server IP address and application port that you bind to the virtual server. However, when you disable port translation, the ServerIron does not perform the translation for the application port. Instead, the ServerIron translates the destination IP address in the client's request to the IP address of a real server, but leaves the application port in the client's request untranslated.

### CLI Example

Here are the commands for implementing the configuration shown in Figure 6.16.

```
ServerIron(config)# server real-name r1 10.0.1.5
ServerIron(config-rs-r1)# port http
ServerIron(config-rs-r1)# port 180
ServerIron(config-rs-r1)# exit
ServerIron(config)# server real-name r2 10.0.2.200
ServerIron(config-rs-r2)# port http
ServerIron(config-rs-r2)# port 180
ServerIron(config-rs-r2)# exit
ServerIron(config)# server virtual-name VIP1 209.157.22.88
ServerIron(config-vs-VIP1)# port http
ServerIron(config-vs-VIP1)# bind http r1 http r2 http
ServerIron(config-vs-VIP1)# exit
ServerIron(config)# server virtual-name VIP2 209.157.22.99
ServerIron(config-vs-VIP2)# port http
ServerIron(config-vs-VIP2)# no port http translate
ServerIron(config-vs-VIP2)# bind http r1 180 r2 180
```

The well-known port (80) is used for VIP1, but an alias (180) is used for VIP2. The real servers actually use port 80 for traffic to both virtual IP addresses. However, the alias port enables the ISP to distinguish among the two IP addresses and their traffic when they display SLB information on the ServerIron. The **no port http translate** command is required. This command enables the ServerIron to send traffic from multiple VIPs to the same real TCP/UDP port on the real server (in this example, "http" (port 80)). If you leave this command out, the ServerIron does not use port 180 as an alias but instead sends the VIP traffic to TCP/UDP port 180 on the real server rather than 80.

---

**NOTE:** Since the untranslated port is logically bound to the translated port and both ports are bound to the same port on the real server, state information for the untranslated port is based on the translated port's state. In this example, state information for port 180 is based on the state for port 80. The state is shown in the **Ms** (Master port state) field of the display produced by the **show server real** command. See "Displaying Real Server Information" on page 6-74.

---

**NOTE:** You can configure the ServerIron to perform health checks on each VIP independently. See "Checking the Health of Multiple Web Sites on the Same Real Server" on page 12-39.

---



To display statistics for the separate real IP addresses, enter the following command at any command prompt:

```
ServerIron(config)# show server real
Server State - 1:enabled, 2:failed, 3:test, 4:suspect, 5:grace_dn, 6:active
Name: r1                IP: 10.0.1.5                : 1 State: 3   Wt: 1   Max-conn: 1
000000
Src-nat (cfg:op) = 0: 0 Dest-nat-(cfg:op) = 0: 0

Port  State  Ms  CurConn  TotConns  Rx-pkts  Tx-pkts  Rx-octet  Tx-octet  Reas
180  enabled 2      0        0        0        0        0        0        0  0
http enabled 0      0        0        0        0        0        0        0  0
Keepalive: Disabled, status code(s) default (200-299, 401)
      HTTP URL: "HEAD /"
defaulunbnd 0      0        0        0        0        0        0        0  0
Server Total      0        0        0        0        0        0        0  0

Name: r2                IP: 10.0.2.200                : 1 State: 3   Wt: 1   Max-conn: 1
000000
Src-nat (cfg:op) = 0: 0 Dest-nat-(cfg:op) = 0: 0

Port  State  Ms  CurConn  TotConns  Rx-pkts  Tx-pkts  Rx-octet  Tx-octet  Reas
http enabled 2      0        0        0        0        0        0        0  0
Keepalive: Disabled, status code(s) default (200-299, 401)
      HTTP URL: "HEAD /"
defaulunbnd 0      0        0        0        0        0        0        0  0
Server Total      0        0        0        0        0        0        0        0  0
```

The lines highlighted in bold indicate the separate HTTP port numbers. The two HTTP lines for real server 1 (r1) indicate that an alias is in use. The first line lists the alias port number and the second line lists the actual port number used by the real server. Even though the same port number is used on the real server, the ServerIron display distinguishes the traffic for the two virtual IP addresses.

---

**NOTE:** The state of the alias HTTP port is always the same as the state of the actual HTTP port used in the packets the ServerIron sends to the real server. The state is shown in the Ms (Master port state) column in the **show server real** display. See "Displaying Real Server Information" on page 6-74.

---

## Web Hosting with Unlimited Virtual IP Addresses

Many ISPs provide subscribers space on their web servers for home pages. Some ISPs provide the user spaces by creating subdirectories off the main domain name of the ISP. For example, an ISP with the domain name "www.budget-web.com" might create directories such as the following for individual subscribers:

- www.budget-web.com/~gillian
- www.budget-web.com/~cindy
- www.budget-web.com/~daisy

Each subscriber's account is on the same IP address. A web user who accesses the server by entering the IP address gains access to the ISP's main page, but then must navigate to the individual subscriber's directory. For home subscribers, this method of web hosting is perfectly satisfactory. However, business subscribers sometimes prefer to have unique domain names.

ISPs that provide separate IP addresses and domain names to their subscribers often do so by configuring multiple IP addresses on their real servers. The real servers have Network Interface Cards (NICs) that support multiple IP addresses. To provide load balancing and redundancy, ISPs sometimes configure multiple real servers with the same contents, but of course with different IP addresses. The ISP configures a unique virtual IP address

(VIP) for each subscriber and uses the ServerIron to map the VIP to real IP addresses on each real server for the subscriber's web site.

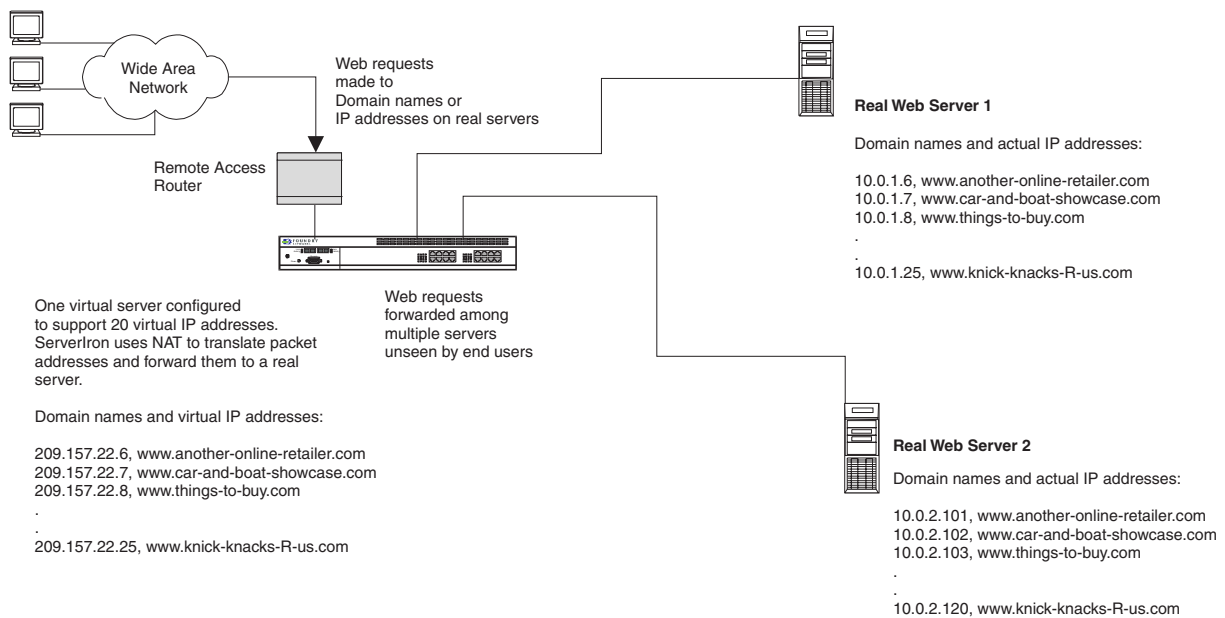
In this type of application, individually configuring a VIP for each subscriber can require a lot of typing. However, the ServerIron makes configuring multiple VIPs easy by allowing you to configure a range of VIPs. When you configure a range, you create a VIP with the lowest address in the range, then specify how many consecutive addresses are in the range. When the ServerIron translates a VIP address configured as part of a host range into its corresponding real IP address on a real server, the ServerIron uses the VIP's offset from the base address to determine the correct real address.

For example, suppose an ISP has two real servers with the following IP address ranges:

- 10.0.1.6 – 10.0.1.25
- 10.0.2.101 – 10.0.2.120

Instead of configuring 20 individual VIPs for these addresses, the ISP administrator can configure one VIP and a host range. In this example, the administrator configures the VIP 209.157.22.6 and adds a host range of 20 addresses to the VIP.

**Figure 6.17 Host range feature used to easily configure a consecutive range of VIPs**



Virtual Domain Name	Virtual IP	TCP Port	Real IP	TCP Port
www.another-online-retailer.com	209.157.22.6	80	S1: 10.0.1.6 S2: 10.0.2.101	80
www.car-and-boat-showcase.com	209.157.22.7	80	S1: 10.0.1.7 S2: 10.0.2.102	80
www.things-to-buy.com	209.157.22.8	80	S1: 10.0.1.8 S2: 10.0.2.103	80
www.knick-knacks-R-us.com	209.157.22.25	80	S1: 10.0.1.25 S2: 10.0.2.120	80

In the example in Figure 6.17, when the ServerIron receives a request for VIP 209.157.22.6, the ServerIron uses the predictor (balancing method) you configured to select one of the real servers, then selects the appropriate IP address on the server. In this case, since 209.157.22.6 is the first address in the VIP range, the ServerIron sends the request to 10.0.1.6 on real server 1 or 10.0.2.101 on real server 2.

---

**NOTE:** To use this feature, make sure the real server has an unbroken range of consecutive IP addresses. Otherwise, you can define separate VIPs and host ranges for each range of unbroken addresses, or you can define a host-range map (see “Host-Range Maps” on page 6-41). Also, when you configure a real server, specify the first address in the host range on that server as that server’s IP address.

---

Suppose the ServerIron receives a request for 209.157.22.8. The ServerIron selects a real server, then applies the offset from the base VIP address to determine the corresponding real server address. In this example, 209.157.22.8 is two addresses higher than the base VIP address. Therefore, when the ServerIron sends the request to a real server, the ServerIron sends the request to a real IP address that is two addresses higher than the base address on the real server. The ServerIron knows to apply the offset because the administrator specified a host range when configuring the virtual server and real servers. The IP address you specify when you configure the real server is the first address in the range.

---

**NOTE:** When health checks are enabled for the ports on the VIPs in a host range, the ServerIron checks the health of the applications on the base IP address only. The ServerIron assumes that the health of an application is the same for all the VIPs within the host range.

---

To configure the ServerIron or switch for this application, enter the following commands:

```
ServerIron(config)# server real-name r1 10.0.1.6
ServerIron(config-rs-r1)# host-range 20
ServerIron(config-rs-r1)# port http
ServerIron(config-rs-r1)# exit
```

These commands configure information for the first real server. The **host-range** command specifies the range of IP addresses the virtual server will represent for the real server. (You do not need to specify the beginning and ending points in the range, just the number of hosts in the range. The IP address you specify for the real server is automatically the first IP address in the range.)

---

**NOTE:** You can specify up to the number of hosts that are available in the sub-net starting with the offset address. For example, if you are configuring a host range on a Class C sub-net and the starting address is 1, then the host range can be up to 255. If the starting address is 100, you can specify up to 155.

---

The **port http** command enables the HTTP port.

The following commands configure information for the second real server.

```
ServerIron(config)# server real-name r2 10.0.2.101
ServerIron(config-rs-r2)# port http
ServerIron(config-rs-r2)# host-range 20
ServerIron(config-rs-r2)# exit
```

After you enter information for the real servers, you are ready to create the virtual server. The following commands create the virtual server.

```
ServerIron(config)# server virtual-name v1 209.157.22.6
ServerIron(config-vs-v1)# host-range 20
ServerIron(config-vs-v1)# port http
ServerIron(config-vs-v1)# bind http r1 http r2 http
ServerIron(config-vs-v1)# exit
```

The **bind** commands associate the http port on each real server with the http port on the virtual server.

---

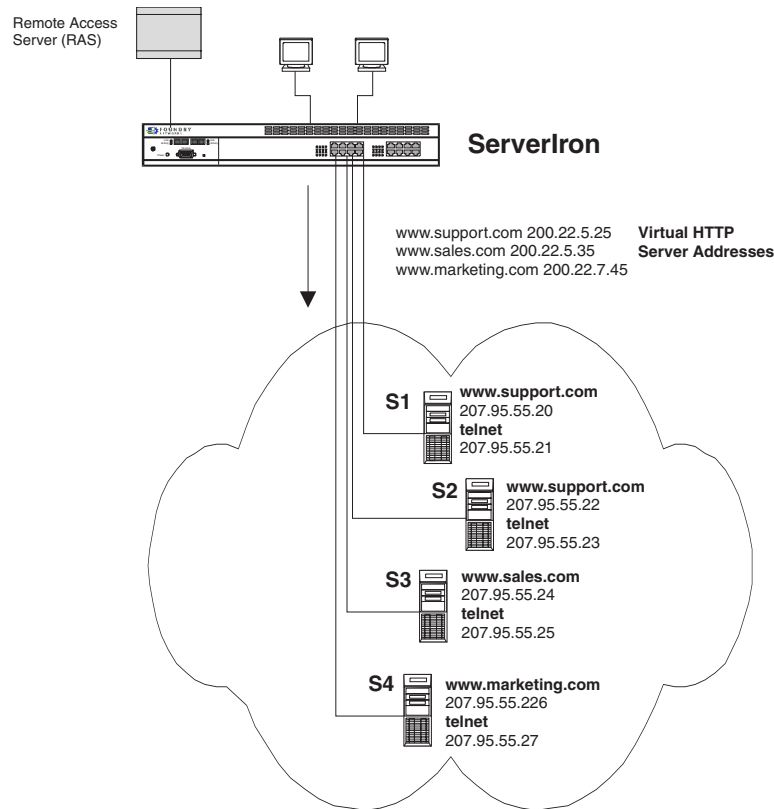
**NOTE:** You also can enter the port number. If you enter the port name, the software uses the well-known number for the port (in this case 80).

---

## SLB Intranet Configuration with HTTP, TELNET Hosting across Multiple Virtual Servers and Multiple Real Servers

A company establishes an Intranet that will handle three different URLs: www.support.com, www.marketing.com, and www.sales.com, as well as Telnet traffic. Telnet traffic is allocated among all four servers, and a server is dedicated to handle each URL with two servers allocated to handle www.support.com requests.

**Figure 6.18 Multiple servers support multiple virtual URLs**



Virtual Domain Name	Virtual IP	TCP Port	Real IP	Port
www.support.com	200.22.3.25	80	S1: 207.95.55.20	80
www.support.com	200.22.3.25	80	S2: 207.95.55.22	80
www.sales.com	200.22.5.35	80	S3: 207.95.55.24	80
www.marketing.com	200.22.7.45	80	S4: 207.95.55.26	80

## TCP/UDP Application Groups

Normally, when the ServerIron selects a real server for a client's request for a TCP/UDP port, there is no guarantee that the ServerIron will select the same real server for subsequent requests from the same client. In many situations, this does not present a problem. Even when the client is requesting the same web page or application, if the content or service is replicated on all the real servers, the client does not know or care which real server provides the content or service for each request.

However, some applications may require that the client continue to use the same real server. For example, an interactive web site might require successive client requests to come to the same server. Other applications might

require that additional TCP/UDP applications also be on the same real server. Some applications may even require the ability to open concurrent connections on the client with different TCP/UDP ports dynamically assigned by the real server.

In all of these cases, the predictor (load-balancing metric) does not ensure that the client returns to the same real server. To accommodate these types of applications, you can configure ports on a virtual server to have the following attributes:

- Sticky connections – When you add a TCP/UDP port to a virtual server, if you specify that the port is “sticky”, a client request for that port always goes to the same real server unless the sticky age timer has expired. The sticky age timer ages out inactive sticky server connections. Possible values are from 2 – 60 minutes. The default is 5 minutes. See “Sticky Age” on page 6-34 for information.
- TCP/UDP application groups (using the **track port** function) – A “primary” TCP/UDP port is grouped with up to four additional TCP/UDP ports. After the ServerIron sends a client request for the primary port to a real server, subsequent requests from the client for ports grouped with the primary port go to the same real server.
- TCP/UDP application groups (using the **track group** function) – Up to eight TCP/UDP ports are grouped together. After the ServerIron sends a client request for any of the grouped ports to a real server, subsequent requests from the client for ports in the group go to the same real server.

---

**NOTE:** You must configure all the ports in a TCP/UDP application group to be “sticky”.

---

- Concurrent connections – The real server can open additional (“concurrent”) TCP/UDP sessions with the client using arbitrary TCP/UDP port numbers.

---

**NOTE:** Although the concurrent connections attribute is similar to application groups, application groups apply to specific TCP/UDP ports that you configure on the virtual server. Concurrent connections enable the real server to arbitrarily determine the TCP/UDP ports and assign them to the client.

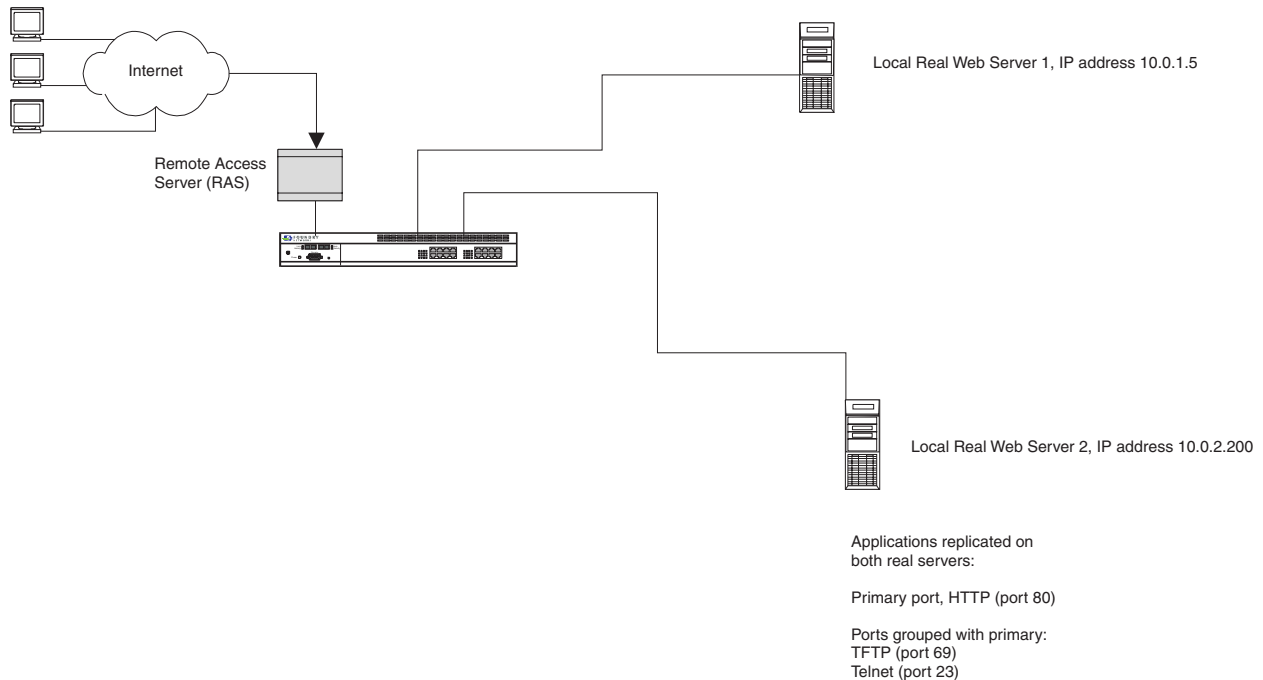
---

---

**NOTE:** For servers that use passive FTP, configure the FTP ports to be both sticky and concurrent.

---

Figure 6.19 shows an example of servers configured with sticky ports and an application group. In this example, the content on each real server is identical. However, some applications on the server require that clients use the same server for subsequent requests to application. The virtual server is configured to make the ports sticky and to group the TFTP and Telnet ports under the HTTP port.

**Figure 6.19 Sticky ports and application group (using the track-port function) used to group TCP/UDP applications**

The following commands show how to implement an application group for this example:

```
ServerIron(config)# server real-name r1 10.0.1.5
ServerIron(config-rs-r1)# port http
ServerIron(config-rs-r1)# port tftp
ServerIron(config-rs-r1)# port telnet
ServerIron(config-rs-r1)# exit
ServerIron(config)# server real-name r2 10.0.2.200
ServerIron(config-rs-r2)# port http
ServerIron(config-rs-r2)# port tftp
ServerIron(config-rs-r2)# port telnet
ServerIron(config-rs-r2)# exit
```

After you enter information for the real servers, you are ready to create the virtual server. The following commands create the virtual server. The **sticky** parameter makes the TCP/UDP ports sticky.

```
ServerIron(config)# server virtual-name v1 209.157.22.1
ServerIron(config-vs-v1)# port 80 sticky
ServerIron(config-vs-v1)# port 69 sticky
ServerIron(config-vs-v1)# port 23 sticky
ServerIron(config-vs-v1)# track 80 69 23
ServerIron(config-vs-v1)# bind 80 r1 80 r2 80
ServerIron(config-vs-v1)# bind 23 r1 23 r2 23
ServerIron(config-vs-v1)# bind 69 r1 69 r2 69
ServerIron(config-vs-v1)# exit
```

The commands above illustrate the track port function. The **track** command groups the Telnet port (23) and the TFTP port (69) under the HTTP port (80); the HTTP port is established as the “primary” port. After the ServerIron sends a client to a real server for the HTTP port, subsequent requests from that client for the HTTP, TFTP, or Telnet port go to the same real server. Up to four ports can be grouped with the primary port.

---

**NOTE:** Since ports 23 and 69 track port 80, state information for the tracking ports (23 and 69 in this example) are based on the tracked port's state (port 80 in this example). The state is shown in the Ms (Master port state) field of the display produced by the **show server real** command. See "Displaying Real Server Information" on page 6-74.

---

The track group function works similarly to the track port function. With the track port function, the client uses the same server for applications associated with the grouped ports, as long as the primary port is active. In contrast, with the track group function, the client uses the same server for applications associated with the grouped ports, as long as **all** the ports in the group are active. After the ServerIron sends a client to a real server for any of the grouped ports, subsequent requests from that client for any of the grouped ports go to the same real server.

The following commands illustrate the track group function:

```
ServerIron(config)# server virtual-name v1 209.157.22.1
ServerIron(config-vs-v1)# port 80 sticky
ServerIron(config-vs-v1)# port 69 sticky
ServerIron(config-vs-v1)# port 23 sticky
ServerIron(config-vs-v1)# track-group 80 69 23
ServerIron(config-vs-v1)# bind 80 r1 80 r2 80
ServerIron(config-vs-v1)# bind 23 r1 23 r2 23
ServerIron(config-vs-v1)# bind 69 r1 69 r2 69
ServerIron(config-vs-v1)# exit
```

In this example, the **track-group** command groups the HTTP port (80), Telnet port (23), and TFTP port (69) together. Whenever a client attempts to connect to a port within the group, the ServerIron ensures all ports in the group are active before granting the connection.

The **sticky** parameter makes the TCP/UDP ports sticky. The sticky parameter must be set for all ports in the group.

After the ServerIron sends a client to a real server for any of these three ports, subsequent requests from that client for the HTTP, TFTP, or Telnet port go to the same real server. Up to eight ports can be grouped together using the track group function. A port can be part of only one group. The **track-group** and **track** commands for a port are mutually exclusive.

## Web Hosting with ServerIron and Real Servers in Different Sub-Nets

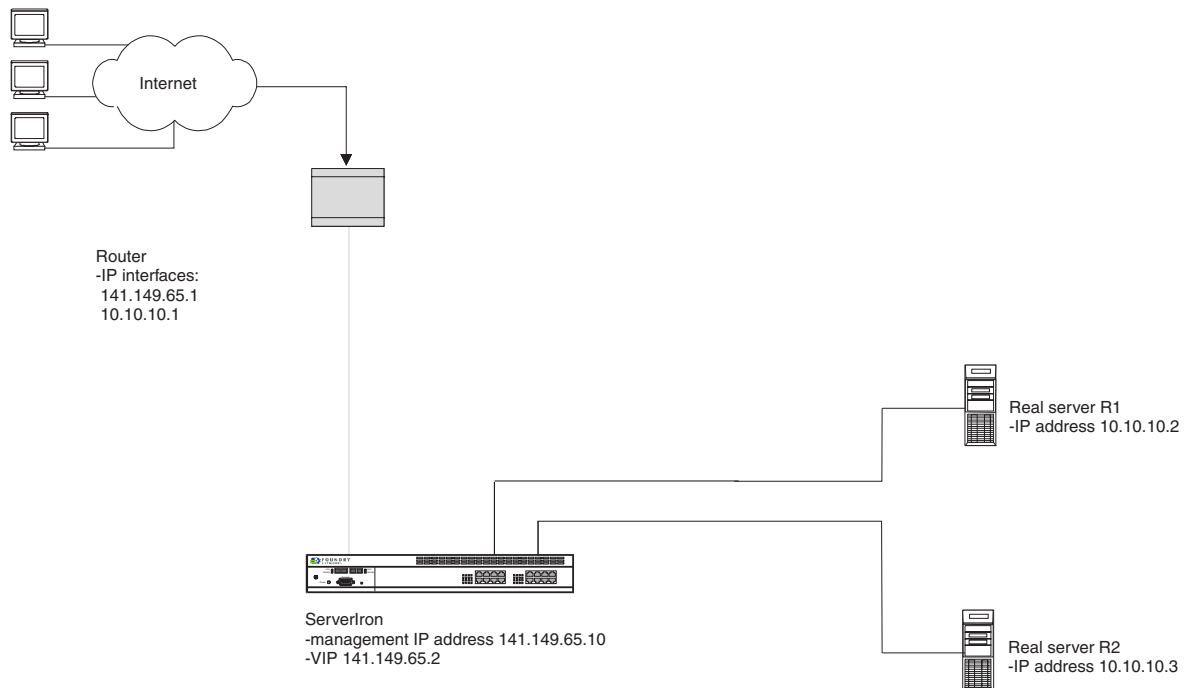
The ServerIron allows you to easily deploy its services in a multinetted environment, without the overhead of configuring routing protocols.

Normally, the ServerIron requires only one IP address, which you use for management access to the device. However, when the ServerIron and real servers are on different sub-nets, you need one of the following:

- Multiple sub-nets configured on the router
- Source NAT enabled and source IP addresses (up to eight) configured on the ServerIron

Figure 6.20 shows an example of a multinetted environment, in which the ServerIron is on one sub-net but the real servers are on another sub-net. The ServerIron is on sub-net 141.149.65.x and the real servers are on sub-net 10.10.10.x.

**Figure 6.20 ServerIron and real servers in multinetted environment – Router configured to route between sub-nets**

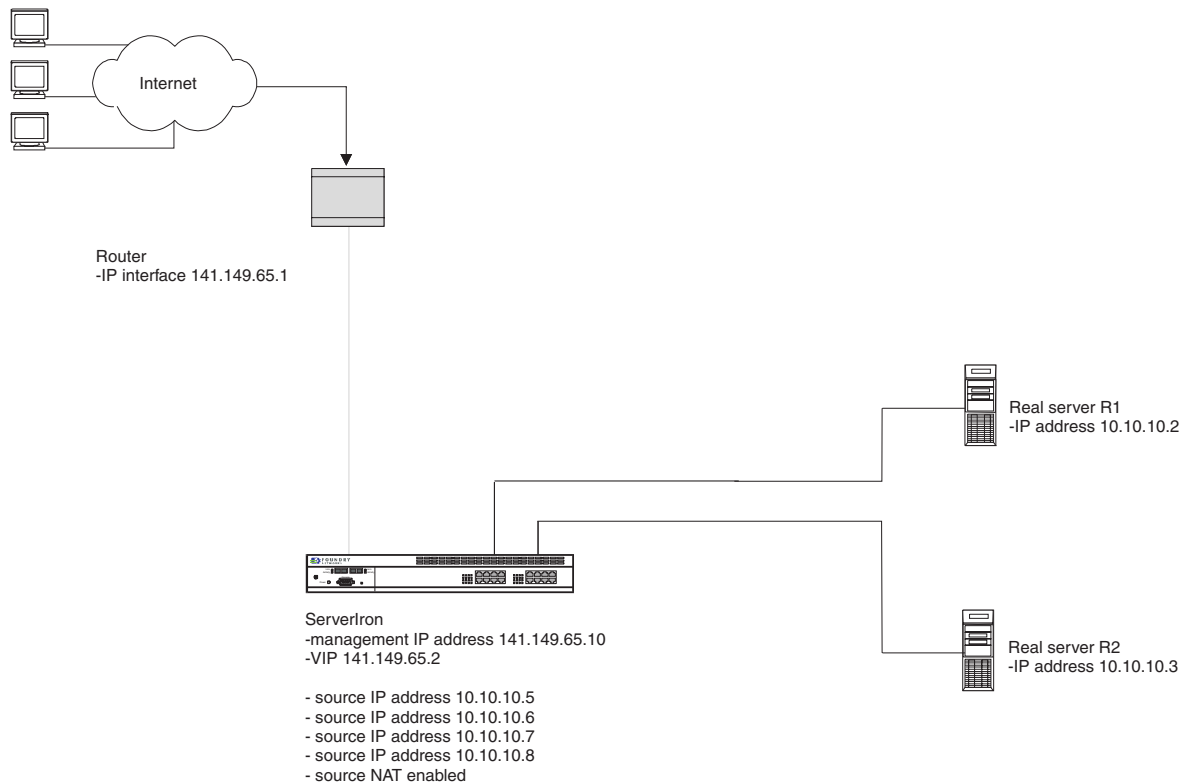


In this example, the ServerIron and the real servers are on different sub-nets, but can communicate because the router is configured with interfaces in both sub-nets. Traffic from the ServerIron to the real servers goes to the router, which routes the traffic to the real servers' sub-net. (The traffic passes back through the ServerIron to reach the real servers, but still must be routed by the router.)

Traffic from the real servers to the ServerIron passes through the ServerIron to the router. The ServerIron acts like a Layer 2 bridge in this case and thus passes the traffic to the router. The router then routes the traffic to the ServerIron's sub-net.

If you have network topology similar to the example in Figure 6.20, but you do not want to configure the router with multiple sub-nets, you can instead enable source NAT and configure a source IP address on the ServerIron. The source IP address allows the ServerIron to be in multiple sub-nets, in addition to the sub-net of the ServerIron's management IP address. Source NAT enables the ServerIron to perform IP address translation on the source address in packets addressed to the real servers. When source NAT is enabled, the ServerIron changes the source address in the IP packets addressed to the real server to the source IP address configured on the ServerIron. Figure 6.21 shows an example of the topology shown in Figure 6.20, but in this case the ServerIron is configured for multiple sub-nets instead of the router.



**Figure 6.21 ServerIron and real servers in multinetted environment – ServerIron configured for source NAT**

In this example, the ServerIron is configured with source IP addresses in the real server's sub-net and source NAT is enabled. The configuration requires five CLI commands or corresponding Web management entries. No reconfiguration of the router is required.

The ServerIron supports a maximum of 64,000 simultaneous connections on each source IP address. This maximum value is based on the architectural limits of IP itself. As a result, if you add only one source IP address, the ServerIron can support up to a maximum of 64,000 simultaneous connections to the real servers. You can configure up to eight source IP addresses, for even more simultaneous connections to the real servers.

Here are the commands for implementing the configuration shown in Figure 6.21.

```
ServerIron(config)# server source-ip 10.10.10.5 255.255.255.0 0.0.0.0
ServerIron(config)# server source-ip 10.10.10.6 255.255.255.0 0.0.0.0
ServerIron(config)# server source-ip 10.10.10.7 255.255.255.0 0.0.0.0
ServerIron(config)# server source-ip 10.10.10.8 255.255.255.0 0.0.0.0
ServerIron(config)# server source-nat

ServerIron(config)# server real-name R1 10.10.10.2
ServerIron(config-rs-r1)# port http
ServerIron(config-rs-r1)# exit

ServerIron(config)# server real-name R2 10.10.10.3
ServerIron(config-rs-r2)# port http
ServerIron(config-rs-r2)# exit

ServerIron(config)# server virtual-name VIP 209.157.22.88
ServerIron(config-vs-VIP1)# port http
ServerIron(config-vs-VIP1)# bind http R1 http R2 http
ServerIron(config-vs-VIP1)# exit
```

---

**NOTE:** If a real server is not reachable from the ServerIron at Layer 2 (does not respond to ARP requests), and if the router connecting the ServerIron to the real server is not running proxy ARP, use the following command instead:

**server remote-name** <name> <ip-addr>

This command adds the server as a remote server. See “Web Hosting with Geographically-Distributed Servers” for information.

Alternatively, enable proxy ARP on the router connecting the ServerIron to the real server.

---

## Web Hosting with Geographically-Distributed Servers

The ServerIron allows you to configure a virtual server to fail over to remote real server IP addresses or VIPs if all local servers become unavailable. The remote servers can be real servers, virtual servers, or a combination of real servers and virtual servers. The remote servers can be locally connected to the ServerIron, connected across a router or even across the Internet.

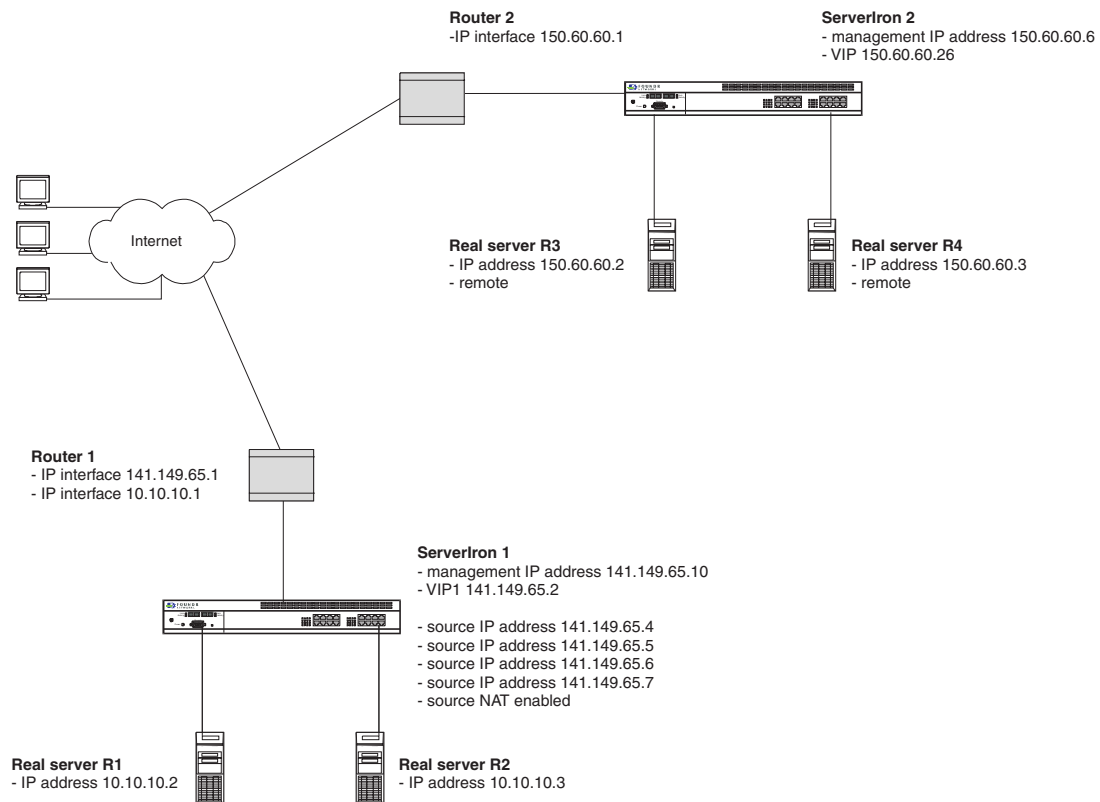
When you configure remote servers in addition to local servers, the ServerIron does not include the remote servers in the predictor (load balancing method). Thus, the remote servers are not used unless all local servers are unavailable.

---

**NOTE:** If you want remote servers to be included in the predictor (load balancing method), configure all the real servers, *including the local ones*, as remote real servers.

---

Figure 6.22 shows an ISP that wants to use load sharing between two local real servers but use remote servers as failovers if both the local real servers are unavailable. The local servers are load balanced by the ServerIron with IP management address 141.149.65.10. The remote servers are load balanced by the ServerIron with IP management address 150.60.60.6. In this example, a VIP on the ServerIron 2 (150.60.60.26) is configured on the ServerIron 1 (141.149.65.10) as a remote server. The remote server can also be a real server’s IP address.

**Figure 6.22 Geographically-distributed servers**

When you configure a ServerIron to fail over to a remote server or to a VIP on another ServerIron, the remote server or VIP typically is on a different sub-net. In this case, the ServerIron must perform some additional address translation to ensure that the traffic from the remote server to the client passes back through the ServerIron that originally serviced the request.

When the ServerIron sends a client request to a local server, it does not change the source IP address of the client's request. However, the ServerIron does change the destination IP address from the VIP requested by the client to the IP address of a real server. When the real server replies to the request, the server's reply is addressed to the client. The ServerIron changes the source IP address of the server's reply to the VIP, then forwards the reply to the client. When the client receives the reply, the reply appears to have come from the VIP.

For the configuration shown in Figure 6.22, you need to enable source NAT. When the ServerIron sends a client request to a real server, the ServerIron does not do source NAT by default. The ServerIron simply performs a destination NAT, changing the VIP address to a real server address. When the real server replies, the ServerIron reverses the destination NAT so that the client sees a reply from the VIP. Real server responses must flow through the ServerIron that performed the original destination NAT so that the NAT can be reversed. Otherwise, the client sees a response from the wrong IP address and either resets the TCP connection or ignores the response.

If you use remote servers in a remote sub-net, you must enable source NAT to force traffic to return to the ServerIron that performed the original destination NAT. The source IP addresses used for source NAT must be in the original ServerIron's broadcast domain. The remote real server replies are addressed to the original ServerIron, not to the client's address. The original ServerIron can then properly reverse the destination NAT.

In Figure 6.22, client requests initially are addressed to VIP1 on ServerIron 1, 141.149.65.2. If the local real servers are healthy, ServerIron 1 distributes traffic to them using destination NAT in the normal way. However, if all the local real servers become unavailable, ServerIron 1 sends traffic to VIP2 on ServerIron 2. ServerIron 1 sends the traffic by using destination NAT in the usual way, translating VIP1's address into VIP2's address. The client's packet is forwarded to the ServerIron's default gateway. By default, if source NAT is not enabled, this is all that happens.

If source NAT is disabled, ServerIron 2 performs a second destination NAT, replacing VIP2's address with R3 or R4's address, depending on which real server is next in the rotation. For this example, assume that ServerIron 2 sends the client request to R3. When R3 replies, the destination address is the client's address and R3's address is replaced by VIP2's address. R3's default gateway forwards this packet directly to the client. ServerIron 1 never sees the packet and never has a chance to reverse the original destination NAT. The client sees a response from 150.60.60.26, rather than 141.149.65.2. The client therefore either resets the TCP connection or simply ignores the response.

To avoid this problem, enable source NAT on ServerIron 1 for VIP2, the remote server. In the example in Figure 6.22, ServerIron 1 has four addresses to use with source NAT:

- 141.149.65.4
- 141.149.65.5
- 141.149.65.6
- 141.149.65.7

When ServerIron 1 sends a packet to VIP2, ServerIron 1 also performs a source NAT using one of these four addresses. The remote servers reply to an address on ServerIron 1 rather than to the client's address. Traffic returns to ServerIron 1 where the original destination NAT is reversed. The client sees a response from VIP1, the same address to which the client sent its request.

All of this is transparent to the client, who simply sends a request to a published IP address and receives a response from that address.

---

**NOTE:** You can enable source NAT globally or on a real server basis (local or remote). If you enable source NAT globally, the ServerIron translates the source address of all client requests. If you enable source NAT locally, on individual real servers, the ServerIron translates the source IP address only for client requests directed to those servers. For example, if you enable source NAT only on the remote servers, the ServerIron translates the source IP addresses only in client requests that the ServerIron directs to the remote servers.

---

Here are the commands for implementing the configuration shown in Figure 6.22. This configuration and all the other configuration information shown here is from the perspective of ServerIron 1. You of course also can configure the remote ServerIron to use a VIP on the local ServerIron as a remote failover.

```
ServerIron-1(config)# server real-name R1 10.10.10.2
ServerIron-1(config-rs-R1)# port http
ServerIron-1(config-rs-R1)# exit

ServerIron-1(config)# server real-name R2 10.10.10.3
ServerIron-1(config-rs-R2)# port http
ServerIron-1(config-rs-R2)# exit
```

The commands shown above configure the local servers. The following commands configure the remote server and enable source NAT for the server. In this example, the remote server is VIP2 configured on ServerIron 2. It is also valid to configure real servers R3 and R4 as the remote servers instead. However, by configuring VIP2 as the remote server, you simplify configuration and also take advantage of the SLB services of ServerIron 2. This example assumes that real servers R3 and R4 and VIP2 are configured on ServerIron 2.

```
ServerIron-1(config)# server remote-name VIP2 150.60.60.26
ServerIron-1(config-rs-VIP2)# source-nat
ServerIron-1(config-rs-VIP2)# port http
ServerIron-1(config-rs-VIP2)# exit
```

The following commands configure VIP1 on ServerIron 1.

```
ServerIron-1(config)# server virtual-name VIP1 141.149.65.2
ServerIron-1(config-vs-VIP1)# port http
ServerIron-1(config-vs-VIP1)# bind http R1 http R2 http VIP2 http
ServerIron-1(config-vs-VIP1)# exit
```

The following **source-ip** commands configure source IP addresses to allow the ServerIron to send a client request to a remote server, receive the response, and then send the response to the client. Notice that the source IP

addresses added to the ServerIron are not in the sub-net of the remote ServerIron. They are in the sub-net that connects the ServerIron's local router to the Internet. The purpose of the source IP addresses in this configuration is to ensure that the responses from remote servers come back to the ServerIron instead of going directly to the clients, so that the ServerIron can properly change the source addresses of the responses back to the VIP requested by the clients.

```
ServerIron-1(config)# server source-ip 141.149.65.4 255.255.255.0 0.0.0.0
ServerIron-1(config)# server source-ip 141.149.65.5 255.255.255.0 0.0.0.0
ServerIron-1(config)# server source-ip 141.149.65.6 255.255.255.0 0.0.0.0
ServerIron-1(config)# server source-ip 141.149.65.7 255.255.255.0 0.0.0.0
```

You can implement this type of configuration using just one source IP address. However, an architectural limitation in IP allows a maximum of 64,000 simultaneous connections on an IP address. As a result, to maximize the number of simultaneous connections the ServerIron can have to the remote VIP, add the maximum number of source IP addresses (eight).

## Using HTTP Redirect with Geographically-Distributed Servers

The application example in the previous section illustrates how to configure the ServerIron to fail over to a remote real server if all local real servers are unavailable. In the previous example, the source NAT feature is used to cause traffic from the remote real server to flow back through the ServerIron to the client.

Depending on the speed of the network connections between the ServerIron and the remote server, you might want the remote server to instead communicate directly with the client. To do so, you can configure a VIP to use HTTP redirect.

Normally, a client expects a response from the VIP and thus regards a TCP SYN ACK (acknowledgment) from the real server as a connection attempt from a different server. If the real server responds directly to the client, the client refuses the real server's TCP SYN ACK. However, you can configure a VIP to use HTTP redirect. In this case, the ServerIron performs address translation as normal when using local real servers. However, if all of the local real servers are unavailable and a remote server is available, the ServerIron sends an HTTP redirect message to the client. The HTTP redirect message instructs the client to redirect its HTTP connection directly to the remote server, bypassing the ServerIron. The client now is talking to the remote server's IP address instead of the VIP.

The remote server can be a real server or another virtual server.

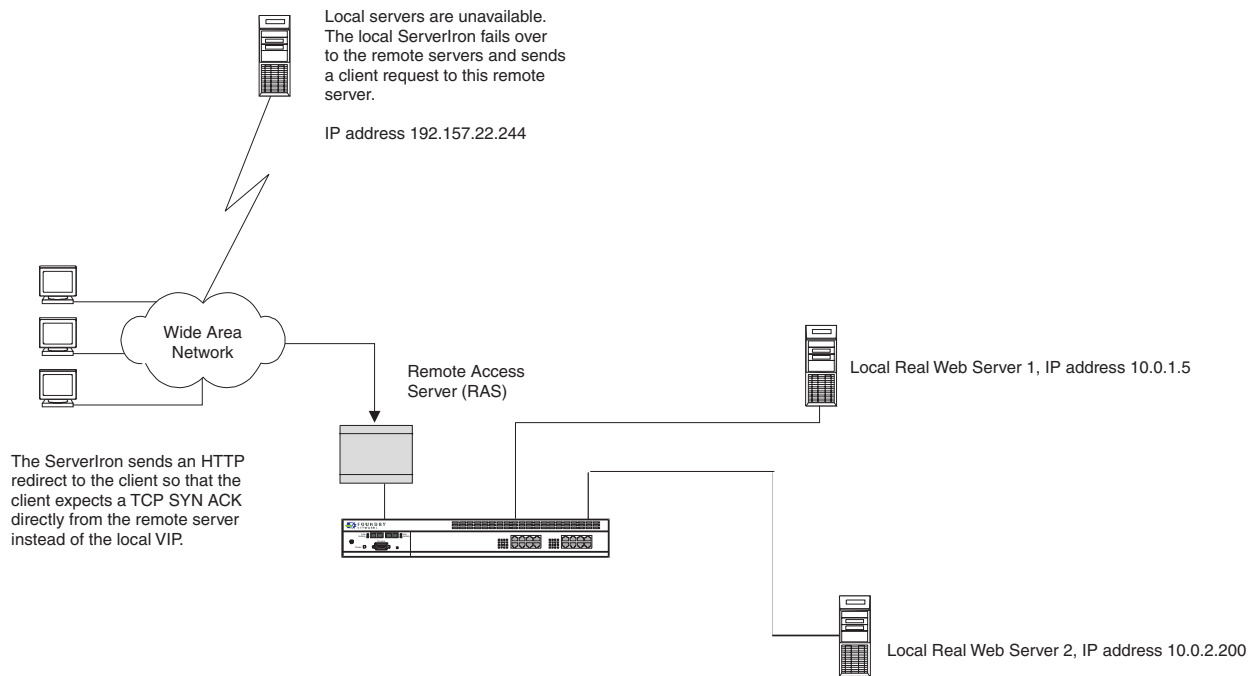
---

**NOTE:** If the user on the client bookmarks a page on the remote server following an HTTP redirect, then uses the bookmark later, the bookmark goes directly to the remote server instead of to the VIP.

---

Figure 6.23 shows an example of HTTP redirect in use.

**Figure 6.23 HTTP redirect used as part of failover to remote server**



Here are the commands for implementing HTTP redirect for the VIP shown in Figure 6.23. The command that enables HTTP redirect is shown in bold.

```
ServerIron(config)# server real-name r1 10.0.1.5
ServerIron(config-rs-r1)# port http
ServerIron(config-rs-r1)# exit

ServerIron(config)# server real-name r2 10.0.2.200
ServerIron(config-rs-r2)# port http
ServerIron(config-rs-r2)# exit

ServerIron(config)# server remote-name r3 192.157.22.244
ServerIron(config-rs-r3)# source-nat
ServerIron(config-rs-r3)# port http
ServerIron(config-rs-r3)# exit

ServerIron(config)# server virtual-name VIP 209.157.22.88
ServerIron(config-vs-VIP1)# port http
ServerIron(config-vs-VIP1)# bind http r1 80 r2 80 r3 80
ServerIron(config-vs-VIP1)# httpredirect
ServerIron(config-vs-VIP1)# exit
```

### Using Reverse Proxy SLB

The Reverse Proxy SLB feature enables you to send client requests for a web page first to a cache server, then to a load balanced real server if the cache server does not have the requested content. This feature is useful for enhancing performance within a load balanced web site for frequently requested web pages.

**NOTE:** You cannot use the Reverse Proxy SLB feature with the TCS cache server spoofing feature on the same ServerIron.

To configure the ServerIron for Reverse Proxy SLB, you configure the real servers and a VIP, then enable Reverse Proxy SLB on the VIP. When the ServerIron receives a request for the VIP, the ServerIron sends the request to a cache server.

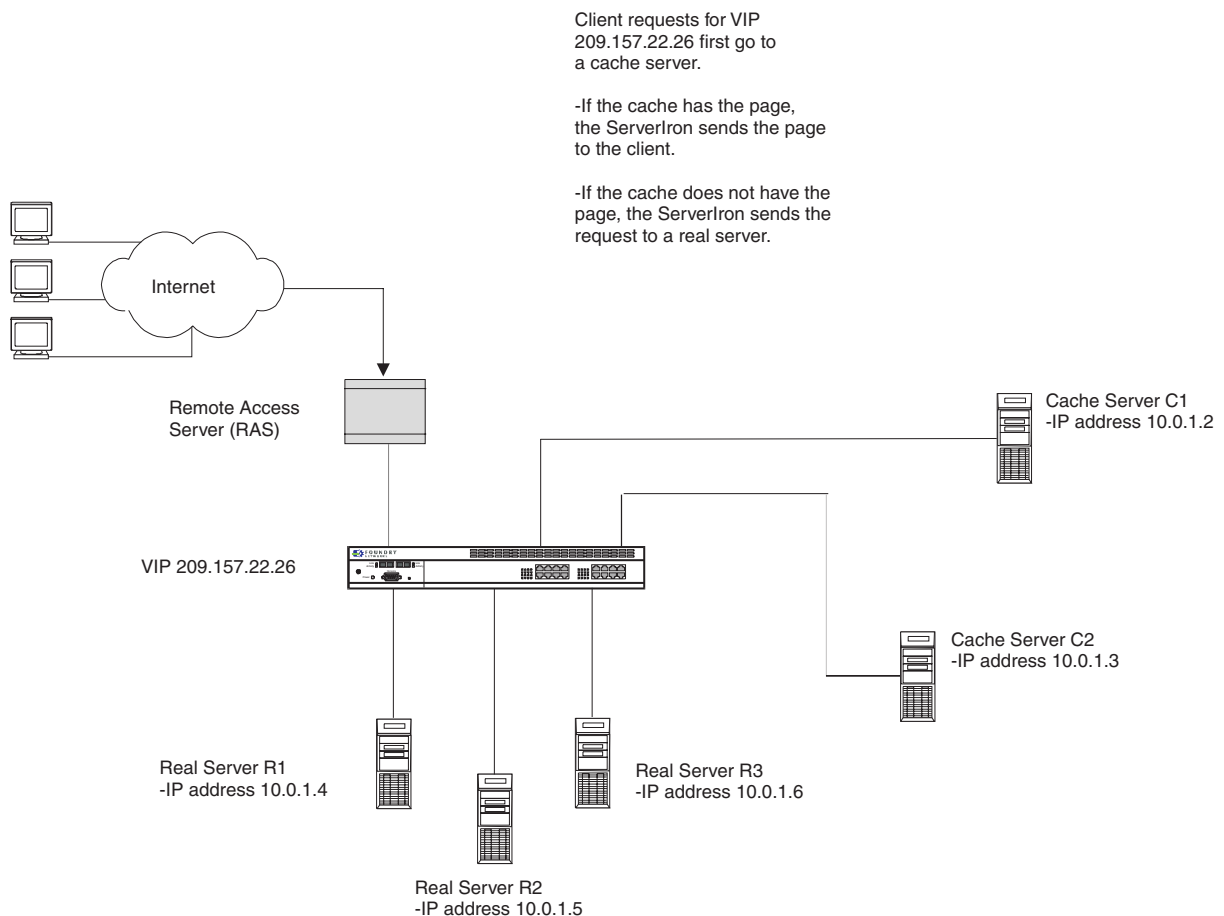
- If the cache server has the requested content, the ServerIron sends the content to the client.
- If the cache server does not have the requested content, the ServerIron redirects the request to a real server. If there is more than one real server, the ServerIron uses the load balancing metric and other SLB parameters you have configured to select the real server.

The ServerIron uses the TCS hash mechanism when selecting a cache server and uses the SLB load balancing method (the predictor) when selecting a real server.

### Basic Example

Figure 6.24 shows an example of a simple Reverse Proxy SLB configuration. Notice that the cache servers and real servers are located close to the web content, as opposed to being located close to the client (or the client's ISP), which is usually the case. Because the cache servers are located close to the content, this type of configuration is sometimes called "reverse caching" or "reverse proxy". The ServerIron is a proxy acting on behalf of the client, but the proxy is located with the web content, rather than with the client's ISP.

**Figure 6.24 Basic Reverse Proxy SLB configuration**



In this example, the ServerIron is configured to send client requests for VIP 209.157.22.26 to a cache server (C1 or C2). If the cache server does not have the requested content, the ServerIron does not send the request to the Internet, as it does in a standard TCS configuration. Instead, the ServerIron sends the request to a load balanced real server.

Here are the CLI commands for implementing the configuration shown above.

The following commands globally enable TCS and configure the cache servers.

```
ServerIron(config)# ip policy 1 cache tcp 80 global
```

```
ServerIron(config)# server cache-name C1 10.0.1.2
ServerIron(config)# server cache-name C2 10.0.1.3
ServerIron(config)# server cache-group 1
ServerIron(config-tc-1)# cache-name C1
ServerIron(config-tc-1)# cache-name C2
ServerIron(config-tc-1)# exit
```

The following commands configure the real servers. Notice that port 80 (HTTP) is added to each server.

```
ServerIron(config)# server real-name R1 10.0.1.4
ServerIron(config-rs-R1)# port 80
ServerIron(config-rs-R1)# exit

ServerIron(config)# server real-name R2 10.0.1.5
ServerIron(config-rs-R2)# port 80
ServerIron(config-rs-R2)# exit

ServerIron(config)# server real-name R3 10.0.1.6
ServerIron(config-rs-R3)# port 80
ServerIron(config-rs-R3)# exit
```

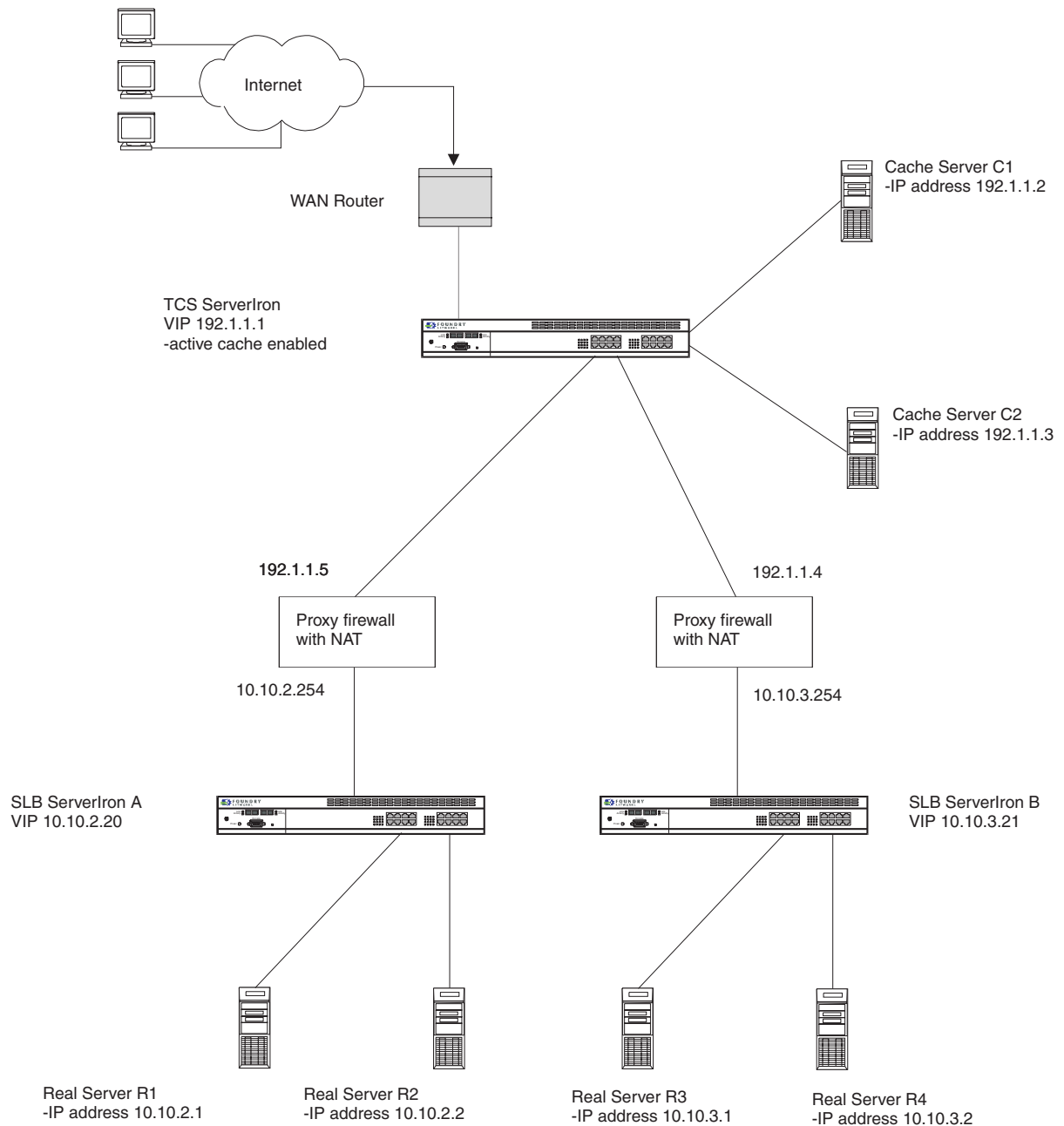
The following commands configure the VIP and save the configuration to the ServerIron's startup-config file on the flash memory. The **cache-enable** command enables the Reverse Proxy SLB feature. You must use this command to use Reverse Proxy SLB. Otherwise, the ServerIron will use the standard TCS behavior, and send client requests to the Internet if the cache server does not have the requested content. The **cache-enable** command configures the ServerIron to send requests that the cache servers cannot fulfill to the real servers instead of to the Internet.

```
ServerIron(config)# server virtual-name VIP1 209.157.22.26
ServerIron(config-vs-VIP1)# port 80
ServerIron(config-vs-VIP1)# bind 80 R1 80 R2 80 R3 80
ServerIron(config-vs-VIP1)# cache-enable
ServerIron(config)# write memory
```

### E-Commerce Example

You can use Reverse Proxy SLB in an E-Commerce environment to offer information that is located on a corporate intranet to the general public without compromising network security. Figure 6.25 shows an example of a Reverse Proxy SLB configuration. Notice that this configuration uses multiple ServerIrons. One of the ServerIrons is configured for TCS and Reverse Proxy SLB while the other two are configured for SLB.



**Figure 6.25 Reverse Proxy SLB configuration in E-Commerce site**

In this example, the cache servers are located in the demilitarized zone (DMZ) of a company. The DMZ is the part of the company's network that is outside the company firewalls but still on the private side of the company's router connection to the Internet.

When a client request comes in from the Internet addressed to VIP 192.1.1.1 on a ServerIron with Reverse Proxy SLB enabled, the ServerIron redirects the request to a cache server. If the cache server has the requested content, the cache server responds directly to the client (through the ServerIron). If the cache server does not have the requested content, the cache server redirects the request to the ServerIron.

Normally, a ServerIron configured for TCS redirects a cache request to the Internet. However, since Reverse Proxy SLB is enabled, the ServerIron instead sends the request to a load balanced real server. In this example, the real servers are firewalls acting as proxy servers. The TCS ServerIron is configured with two real servers.

Each of them is actually a firewall. Each of the firewalls is configured to perform NAT to translate packets addressed to its interface with the ServerIron into the VIP configured on the SLB ServerIron connected to it. Thus, if the TCS ServerIron sends a client request to firewall interface 192.1.1.5 (configured on the TCS ServerIron as a real server), the firewall translates the packet's destination address into VIP 10.10.2.20.

---

**NOTE:** This example assumes that the firewalls are properly configured to perform the address translations needed for this configuration.

---

The ServerIron to which the firewall (proxy server) sends the client request sends the request to a real server, then sends the response back to the firewall, which again performs NAT and sends the response to the cache server. The cache server then sends the requested content to the client. From the client's perspective, the response arrives from IP address 192.1.1.1. This is true whether the content was on the cache server when the client requested it or the cache server needed to obtain the content from a real server before providing it to the client.

#### **Commands on TCS ServerIron**

The following commands configure the TCS ServerIron. Notice that two real servers are added to the ServerIron. These real servers are actually the firewalls. The real server IP addresses are the firewall interfaces with the TCS ServerIron. Also notice that the ports on the VIP are bound to the real servers, as in a standard TCS configuration.

```
ServerIron(config)# ip policy 1 cache tcp 80 global
ServerIron(config)# server cache-name C1 192.1.1.2
ServerIron(config)# server cache-name C2 192.1.1.3

ServerIron(config)# server cache-group 1
ServerIron(config-tc-1)# cache-name C1
ServerIron(config-tc-1)# cache-name C2
ServerIron(config-tc-1)# exit

ServerIron(config)# server real-name Proxy1 192.1.1.5
ServerIron(config-rs-Proxy1)# port 80
ServerIron(config-rs-Proxy1)# port 443
ServerIron(config-rs-Proxy1)# exit

ServerIron(config)# server real-name Proxy2 192.1.1.4
ServerIron(config-rs-Proxy2)# port 80
ServerIron(config-rs-Proxy2)# port 443
ServerIron(config-rs-Proxy2)# exit

ServerIron(config)# server virtual-name VIP1 192.1.1.1
ServerIron(config-vs-VIP1)# port 80
ServerIron(config-vs-VIP1)# port 443
ServerIron(config-vs-VIP1)# bind 80 Proxy1 80 Proxy2 80
ServerIron(config-vs-VIP1)# bind 443 Proxy1 443 Proxy2 443
ServerIron(config-vs-VIP1)# cache-enable
ServerIron(config-vs-VIP1)# exit
ServerIron(config)# write memory
```

#### **Commands on SLB ServerIron A**

```
ServerIron(config)# server real-name R1 10.10.2.1
ServerIron(config-rs-R1)# port 80
ServerIron(config-rs-R1)# port 443
ServerIron(config-rs-R1)# exit

ServerIron(config)# server real-name R2 10.10.2.2
ServerIron(config-rs-R2)# port 80
ServerIron(config-rs-R2)# port 443
ServerIron(config-rs-R2)# exit

ServerIron(config)# server virtual-name VIP2 10.10.2.20
ServerIron(config-vs-VIP2)# port 80
ServerIron(config-vs-VIP2)# port 443
```

```
ServerIron(config-vs-VIP2)# bind 80 R1 80 R2 80
ServerIron(config-vs-VIP2)# bind 443 R1 443 R2 443
ServerIron(config-vs-VIP2)# exit
ServerIron(config)# write memory
```

#### **Commands on SLB ServerIron B**

```
ServerIron(config)# server real-name R3 10.10.3.1
ServerIron(config-rs-R3)# port 80
ServerIron(config-rs-R3)# port 443
ServerIron(config-rs-R3)# exit

ServerIron(config)# server real-name R4 10.10.3.2
ServerIron(config-rs-R4)# port 80
ServerIron(config-rs-R4)# port 443
ServerIron(config-rs-R4)# exit

ServerIron(config)# server virtual-name VIP3 10.10.3.21
ServerIron(config-vs-VIP2)# port 80
ServerIron(config-vs-VIP2)# port 443
ServerIron(config-vs-VIP2)# bind 80 R3 80 R4 80
ServerIron(config-vs-VIP2)# bind 443 R3 443 R4 443
ServerIron(config-vs-VIP2)# exit
ServerIron(config)# write memory
```

### **Load Balancing Streaming Media Files**

The ServerIron can perform load balancing for the following kinds of streaming media files:

- VDOLive – TCP port 7000
- StreamWorks – UDP port 1558
- Microsoft Media Service – TCP port 1755
- Real Networks' Real Audio/Video – TCP port 7070
- Microsoft VxTreme – TCP port 12468
- Real Networks' RealMedia – TCP port 554
- Apple's QuickTime – TCP port 554

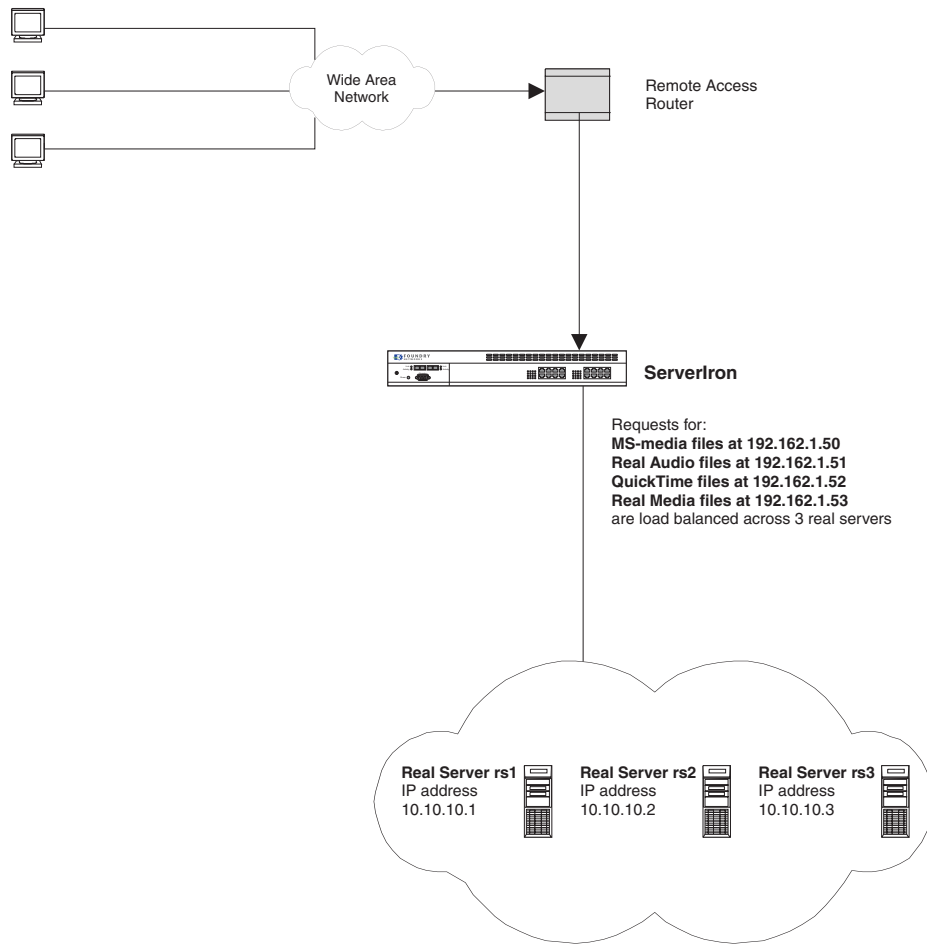
---

**NOTE:** Some streaming media types can use ports other than their well-known port. However, the ServerIron generally supports only the well-known port. For example, QuickTime can use port 7070, in addition to the more common 554. The ServerIron currently supports streaming media load balancing for QuickTime only on port 554.

---

Figure 6.26 shows a sample configuration where requests for three kinds of streaming media files are load balanced across three real servers.

**Figure 6.26 Streaming Media SLB configuration**



Virtual IP	Predictor	TCP Port	Real IP	TCP Port
192.162.1.50	weighted	1755	rs1: 10.10.10.1 rs2: 10.10.10.2 rs3: 10.10.10.3	1755
192.162.1.51	least-conn	7070	rs1: 10.10.10.1 rs2: 10.10.10.2 rs3: 10.10.10.3	7070
192.162.1.52	round-robin	554	rs1: 10.10.10.1 rs2: 10.10.10.2 rs3: 10.10.10.3	554

In this configuration, all the streaming media content is located on the three real servers. Requests for MS-media files on VIP 192.162.1.50 are load balanced across the real servers using the weighted predictor; requests for Real Audio files on VIP 192.162.1.51 are load balanced using the least connections predictor; and requests for QuickTime files on VIP 192.162.1.52 are load balanced using the round-robin predictor.

The following commands configure the real servers in Figure 6.26.

```
ServerIron(config)# server real-name rs1 10.10.10.1
```

```

ServerIron(config-rs-rs1)# port rtsp
ServerIron(config-rs-rs1)# port pnm
ServerIron(config-rs-rs1)# port mms
ServerIron(config-rs-rs1)# exit

ServerIron(config)# server real-name rs2 10.10.10.2
ServerIron(config-rs-rs2)# port rtsp
ServerIron(config-rs-rs2)# port pnm
ServerIron(config-rs-rs2)# port mms
ServerIron(config-rs-rs2)# exit

ServerIron(config)# server real-name rs3 10.10.10.3
ServerIron(config-rs-rs3)# port rtsp
ServerIron(config-rs-rs3)# port pnm
ServerIron(config-rs-rs3)# port mms
ServerIron(config-rs-rs3)# exit

```

The following commands bind the real servers to the virtual servers in Figure 6.26.

```

ServerIron(config)# server virtual-name MSmedia1755 192.162.1.50
ServerIron(config-vs-MSmedia1755)# predictor weighted
ServerIron(config-vs-MSmedia1755)# port mms
ServerIron(config-vs-MSmedia1755)# bind mms rs1 mms rs2 mms rs3 mms
ServerIron(config-vs-MSmedia1755)# exit

ServerIron(config)# server virtual-name real7070 192.162.1.51
ServerIron(config-vs-real7070)# predictor least-conn
ServerIron(config-vs-real7070)# port pnm
ServerIron(config-vs-real7070)# bind pnm rs1 pnm rs2 pnm rs3 pnm
ServerIron(config-vs-real7070)# exit

ServerIron(config)# server virtual-name quicktime554 192.162.1.52
ServerIron(config-vs-quicktime554)# predictor round-robin
ServerIron(config-vs-quicktime554)# port rtsp
ServerIron(config-vs-quicktime554)# bind rtsp rs1 rtsp rs2 rtsp rs3 rtsp
ServerIron(config-vs-quicktime554)# exit

```

---

**NOTE:** The ServerIron supports configurations that use port 80 for streaming media. However, a Layer 7 health check may fail because a status code of 404 can be returned in response to GET or HEAD requests. To work around this, you must configure the health check so that 404 is an acceptable status code. To do this, use the command **port http status-code 200 404** in the real server configuration.

---

## Globally Disabling TCP or UDP Ports

You can globally disable a Layer 4 port on the ServerIron. The port can be disabled for all real servers, all virtual servers or all real and virtual servers. After you disable a port globally, you can enable the port on individual real or virtual servers as necessary. By default, all real and virtual ports are enabled.

When the ServerIron is booted, if the command to globally disable a real or virtual port exists in the startup-config file, the specified port is disabled at startup. When a real or virtual port is created, and the port has been disabled globally, the real or virtual port is disabled as well. You must enable the port explicitly.

To disable all real HTTP ports:

```

ServerIron(config)# server port 80
ServerIron(config-port-http)# disable real
ServerIron(config-port-http)#

```

To disable all virtual HTTP ports:

```

ServerIron(config)# server port 80
ServerIron(config-port-http)# disable virtual
ServerIron(config-port-http)#

```

To disable all real and virtual HTTP ports:

```
ServerIron(config)# server port 80
ServerIron(config-port-http)# disable
ServerIron(config-port-http)#
```

**Syntax:** disable [real | virtual]

---

# Chapter 7

## Configuring Symmetric SLB and SwitchBack

This chapter describes how to configure Symmetric SLB (SSLB) and SwitchBack for Server Load Balancing (SLB).

- Symmetric SLB enhances performance by allowing you to use a pair of ServerIrons to actively load balance VIPs, while at the same time providing mutual backup on an individual VIP basis.
- SwitchBack configurations can enhance server response time and increase capacity on the ServerIron, by allowing server responses to bypass the ServerIron on the way to clients and at the same time to increasing the number of simultaneous connections the ServerIron can support.

### Using Symmetric Server Load Balancing

The Symmetric Server Load Balancing (SLB) feature increases performance and capacity and simplifies redundant topology. The feature provides these benefits by allowing you to implement redundancy on an individual VIP basis. Unlike the conventional hot-standby configuration, you can actively use all the ServerIrons in a Symmetric SLB configuration at the same time.

You can configure the following types of Symmetric SLB:

- Active-Standby – Each ServerIron is the “active” ServerIron for a specific set of VIPs, while the other ServerIrons are backups for those VIPs.
- Active-Active – Both ServerIrons in the SSLB configuration to be active for the same application port and VIP, at the same time. See “Active-Active SSLB” on page 7-8.

---

**NOTE:** Active-Active Symmetric SLB is supported only on the ServerIron 400 and ServerIron 800.

---

### Active-Standby SSLB

In Active-Standby SSLB, you determine which ServerIrons are active and backup for a VIP by associating a priority with the VIP when you configure it on a given ServerIron. You assign a different priority to the same VIP on each ServerIron. The ServerIron on which the VIP has the highest priority is the default “active” ServerIron for that VIP and the others are standbys. When all ServerIrons and associated links are available, the ServerIron with the highest priority for the VIP services the VIP. The other ServerIrons are hot standbys for the VIP.

You can assign each VIP a priority from 0 – 255. You can configure up to 255 ServerIrons in a Symmetric SLB configuration. You also can configure the priority to dynamically adjust to changes in the health of applications on the VIP.

Symmetric SLB does not require any changes to the Spanning Tree configuration in the network. Regardless of whether the network is using Spanning Tree, Symmetric SLB provides redundancy for the VIPs and allows all the ServerIrons configured for Symmetric SLB to actively perform Server Load Balancing.

In addition, you do not need to dedicate ServerIron links to Symmetric SLB. Symmetric SLB works within the network's topology.

**NOTE:** You cannot have a router hop between the ServerIrons. They must have Layer 2 connectivity.

**NOTE:** You cannot use Hot Standby and Symmetric SLB features on the same ServerIron.

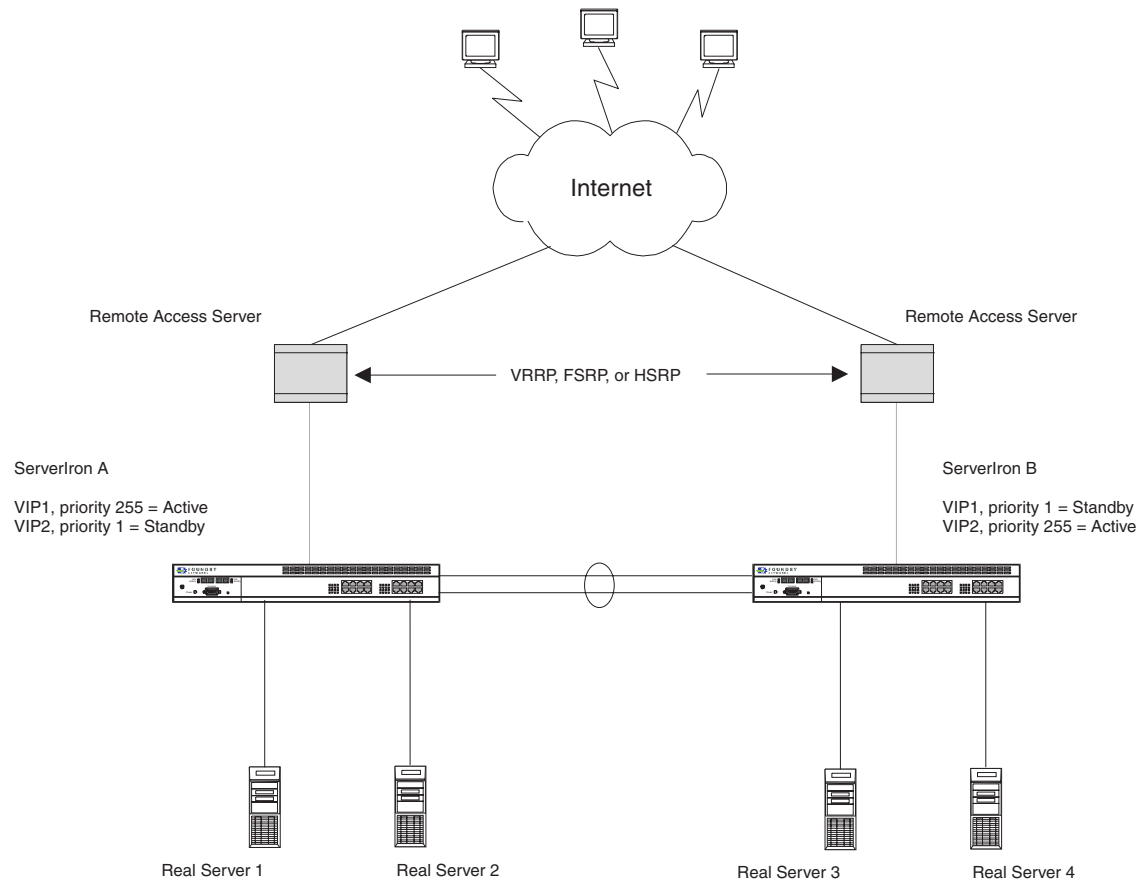
Figure 7.1 shows an example of a Symmetric SLB configuration. In this example, two ServerIrons are each connected to two real servers. For simplicity, this example uses two VIPs whose contents are replicated on all four servers. To implement the Symmetric SLB configuration shown in Figure 7.1, you configure both VIPs on each ServerIron. In this example, real servers 1 and 2 are bound to VIPs 1 – 2 configured on ServerIron A and real servers 3 and 4 are bound to VIPs 1 – 2 configured on ServerIron B.

Because multiple VIPs are mapping to the same ports on the same real servers, TCP/UDP port binding is used. Thus, port 180 on VIP2 on ServerIron A and on VIP1 on ServerIron 2 is a logical port that is bound to port 80 on the real servers. For more information, see “Many-To-One TCP/UDP Port Binding” on page 6-98.

To load balance the VIPs across the ServerIrons, you give one of the VIPs a higher priority on the first ServerIron and give the second VIP a higher priority on the second ServerIron, as shown in the following table and in Figure 7.1. Both ServerIrons are actively servicing traffic, but each VIP is serviced by only one ServerIron.

ServerIron	Domain Name	Virtual IP (VIP) Address	Priority	VIP's TCP Port	Real IP Address	Real Server's TCP Port
A	www.abc.com	VIP1: 209.157.22.100	254	80	Real Server 1: 10.0.0.1  Real Server 2: 10.0.0.2	80  80
A	www.def.com	VIP2: 209.157.22.101	2	80	Real Server 1: 10.0.1.1  Real Server 2: 10.0.1.2	180  180
B	www.abc.com	VIP1: 209.157.22.100	2	80	Real Server 3: 10.0.0.1  Real Server 4: 10.0.0.2	180  180
B	www.def.com	VIP2: 209.157.22.101	254	80	Real Server 3: 10.0.1.1  Real Server 4: 10.0.1.2	80  80



**Figure 7.1 Symmetric SLB**

**NOTE:** The trunk link between the two ServerIrons provides Layer 2 connectivity but the links are not required exclusively for Symmetric SLB. The links are standard Layer 2 links and can thus be used for normal Layer 2 traffic. Symmetric SLB communication between the ServerIrons uses the same Layer 2 paths as other Layer 2 traffic. Figure 7.4 shows an example of a Symmetric SLB configuration in which the ServerIrons are not directly connected but are instead connected through another Layer 2 switch.

When both ServerIrons and all links are available, client traffic for VIP1 is always serviced by ServerIron A, because that VIP has a higher priority on ServerIron A than it does on ServerIron B. Likewise, VIP2 is always serviced by ServerIron B. This is illustrated in Figure 7.2.

**Figure 7.2 VIP priorities determine which ServerIron services them**

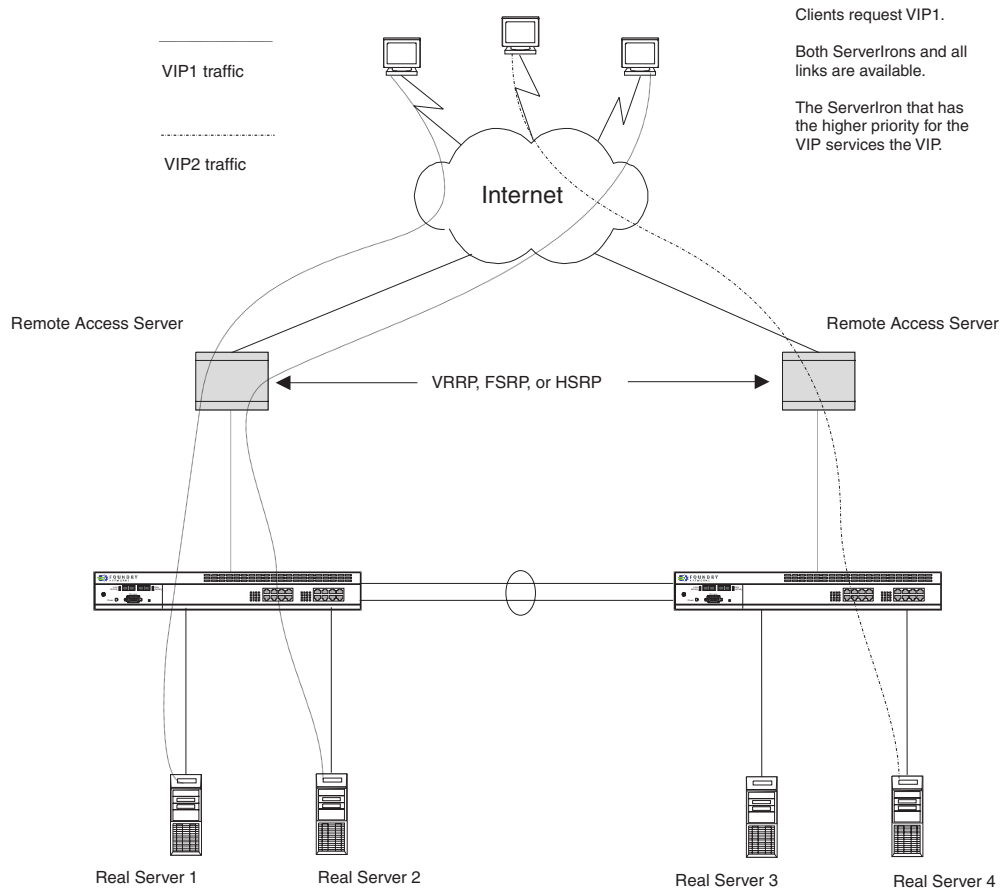
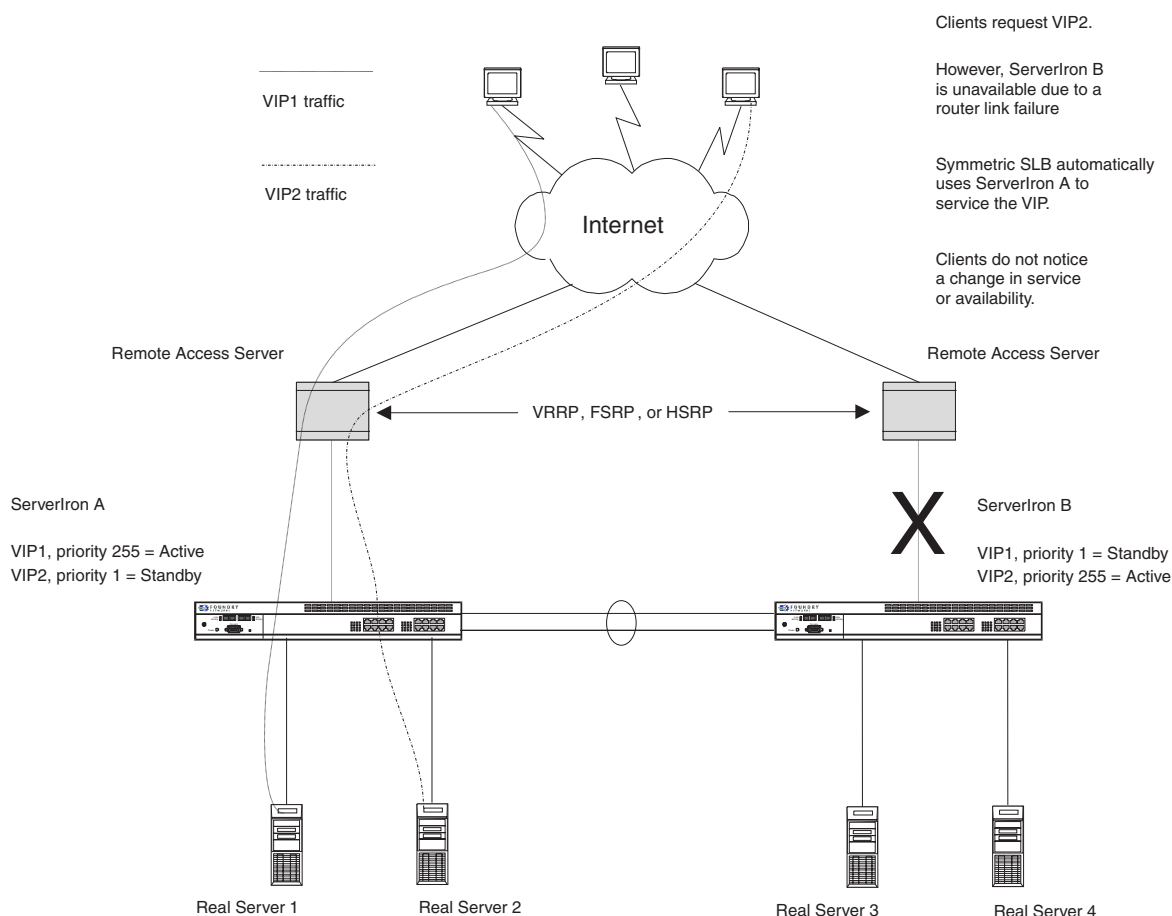


Figure 7.3 shows a link failure on the Remote Access Server (RAS) connecting the clients to ServerIron B makes ServerIron B temporarily unavailable. Symmetric SLB compensates for the unavailable ServerIron by servicing VIP2 as well as continuing to service VIP1.

**Figure 7.3 Symmetric SLB automatically compensates for unavailable equipment and links**

When ServerIron B becomes available again, Symmetric SLB automatically restores the active and standby relationships to the way they were before, based on the priorities assigned to the VIPs.

### CLI Commands

To implement the configuration shown in Figure 7.1, enter the following commands. Notice that the commands are very similar on each ServerIron. The only difference is the value specified with the **sym-priority** command.

These examples show statically configured priorities. You can configure the priorities to automatically adjust to changes in the health of applications on the VIP. See “Using Dynamic SSLB Priority” on page 7-10.

**NOTE:** You must configure all the real servers on both ServerIrons, and bind the VIPs on each ServerIron to all the real servers.

### Commands on ServerIron A

Enter the following commands to configure the real servers. You must configure all the real servers on both ServerIrons.

Notice that two HTTP ports are added to each real server. This type of configuration requires that you use the TCP/UDP port binding feature to bind the ports on the two real servers to the same port on the virtual server. For information, see “Many-To-One TCP/UDP Port Binding” on page 6-98.

```
ServerIronA(config)# server real-name Real_Server_1 10.0.0.1
ServerIronA(config-rs-Real_Server_1)# port http
ServerIronA(config-rs-Real_Server_1)# port 180
ServerIronA(config-rs-Real_Server_1)# exit
```

```

ServerIronA(config)# server real-name Real_Server_2 10.0.0.2
ServerIronA(config-rs-Real_Server_2)# port http
ServerIronA(config-rs-Real_Server_2)# port 180
ServerIronA(config-rs-Real_Server_2)# exit
ServerIronA(config)# server real-name Real_Server_3 10.0.1.1
ServerIronA(config-rs-Real_Server_3)# port http
ServerIronA(config-rs-Real_Server_3)# port 180
ServerIronA(config-rs-Real_Server_3)# exit
ServerIronA(config)# server real-name Real_Server_4 10.0.1.2
ServerIronA(config-rs-Real_Server_4)# port http
ServerIronA(config-rs-Real_Server_4)# port 180
ServerIronA(config-rs-Real_Server_4)# exit

```

Enter the following commands to configure the VIPs. Make sure you bind all the real servers to each VIP.

Notice that the **sym-priority** command is entered for each VIP to set the VIP's Symmetric SLB priority. The priority determines which ServerIron is active for that VIP. A higher priority is favored over a lower priority. Thus, in this example VIP1's priority on ServerIron A (254) is higher than on ServerIron B (2). VIP2's priority is higher on ServerIron B (254) than on ServerIron A (2). You can specify any value from 0 – 255 for the priority but do not specify the same priority for the same VIP on multiple ServerIrons. The lines containing the **sym-priority** command are shown in bold type.

---

**NOTE:** If you specify 0, the CLI removes the priority. When you save the configuration to the startup-config file, the **sym-priority** command is removed. Use this method to remove the priority from a VIP. You cannot remove the priority using the **no sym-priority** command.

---

```

ServerIronA(config)# server virtual-name VIP1 209.157.22.100
ServerIronA(config-vs-VIP1)# port http
ServerIronA(config-vs-VIP1)# bind http Real_Server_1 http Real_Server_2 http
Real_Server_3 http Real_Server_4 http
ServerIronA(config-vs-VIP1)# sym-priority 254
ServerIronA(config-vs-VIP1)# exit
ServerIronA(config)# server virtual-name VIP2 209.157.22.101
ServerIronA(config-vs-VIP2)# port http
ServerIronA(config-vs-VIP2)# bind http Real_Server_1 180 Real_Server_2 180
Real_Server_3 180 Real_Server_4 180
ServerIronA(config-vs-VIP2)# no port http translate
ServerIronA(config-vs-VIP2)# sym-priority 2
ServerIronA(config-vs-VIP2)# exit
ServerIronA(config)# write memory

```

### Commands on ServerIron B

Enter the following commands to configure the real servers.

```

ServerIronB(config)# server real-name Real_Server_1 10.0.0.1
ServerIronB(config-rs-Real_Server_1)# port http
ServerIronB(config-rs-Real_Server_1)# port 180
ServerIronB(config-rs-Real_Server_1)# exit
ServerIronB(config)# server real-name Real_Server_2 10.0.0.2
ServerIronB(config-rs-Real_Server_2)# port http
ServerIronB(config-rs-Real_Server_2)# port 180
ServerIronB(config-rs-Real_Server_2)# exit
ServerIronB(config)# server real-name Real_Server_3 10.0.1.1
ServerIronB(config-rs-Real_Server_3)# port http
ServerIronB(config-rs-Real_Server_3)# port 180
ServerIronB(config-rs-Real_Server_3)# exit
ServerIronB(config)# server real-name Real_Server_4 10.0.1.2
ServerIronB(config-rs-Real_Server_4)# port http
ServerIronB(config-rs-Real_Server_4)# port 180
ServerIronB(config-rs-Real_Server_4)# exit

```

Enter the following commands to configure the VIPs. The lines containing the **sym-priority** command are shown in bold type.

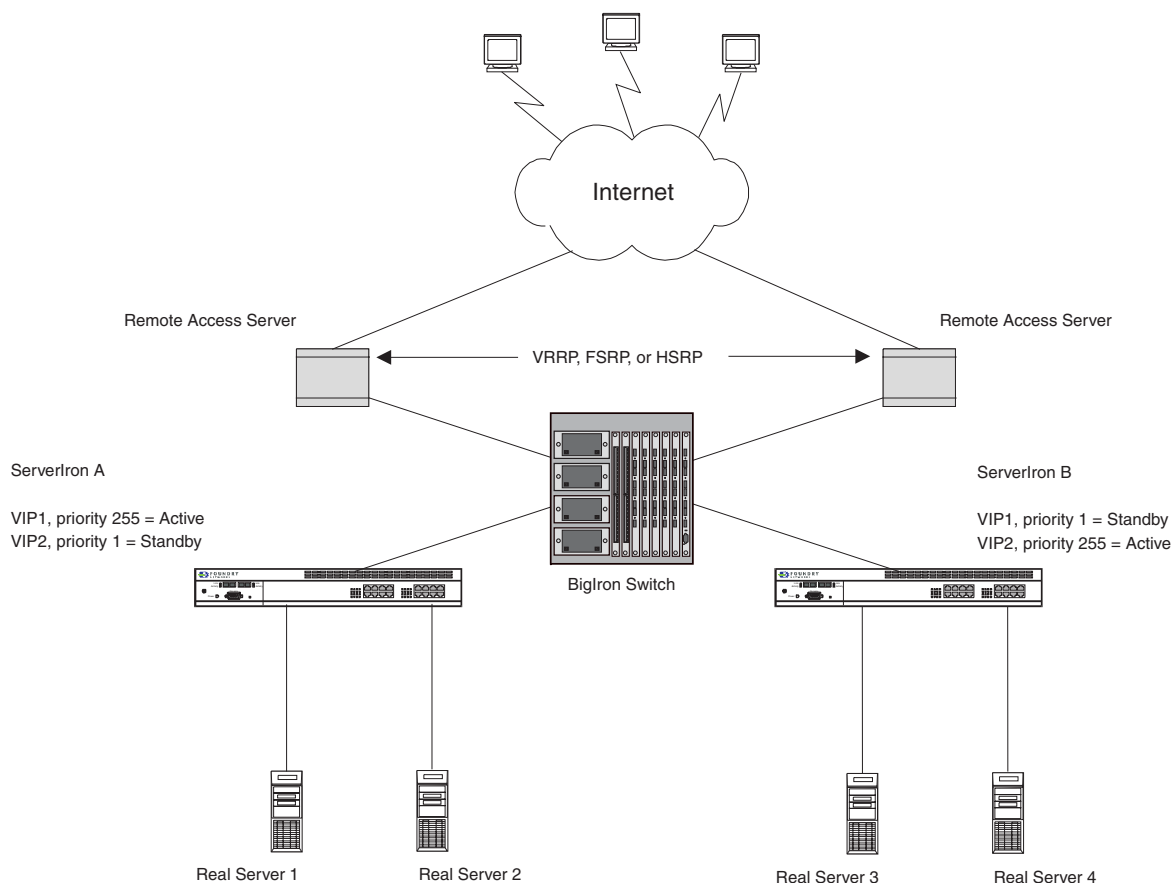
```
ServerIronB(config)# server virtual-name VIP1 209.157.22.100
ServerIronB(config-vs-VIP1)# port http
ServerIronB(config-vs-VIP1)# bind http Real_Server_1 180 Real_Server_2 180
Real_Server_3 180 Real_Server_4 180
ServerIronB(config-vs-VIP2)# no port http translate
ServerIronB(config-vs-VIP1)# sym-priority 2
ServerIronB(config-vs-VIP1)# exit
ServerIronB(config)# server virtual-name VIP2 209.157.22.101
ServerIronB(config-vs-VIP2)# port http
ServerIronB(config-vs-VIP2)# bind http Real_Server_1 http Real_Server_2 http
Real_Server_3 http Real_Server_4 http
ServerIronB(config-vs-VIP2)# sym-priority 254
ServerIronB(config-vs-VIP2)# exit
ServerIronB(config)# write memory
```

### Example Without Direct Links Between the ServerIrons

The Symmetric SLB configuration shown in Figure 7.1 shows a direct link between the two ServerIrons. The link provides standard Layer 2 connectivity and is not a dedicated link required exclusively for Symmetric SLB. Figure 7.4 shows a configuration that is similar to the one shown in Figure 7.1, except that the Layer 2 connectivity between the ServerIrons is provided by a Layer 2 switch instead of a direct link between the ServerIrons.

In terms of Symmetric SLB operation, this configuration is equivalent to the configuration in Figure 7.1. The CLI commands for implementing the configuration on the two ServerIrons are also the same.

**Figure 7.4 Another Symmetric SLB configuration**



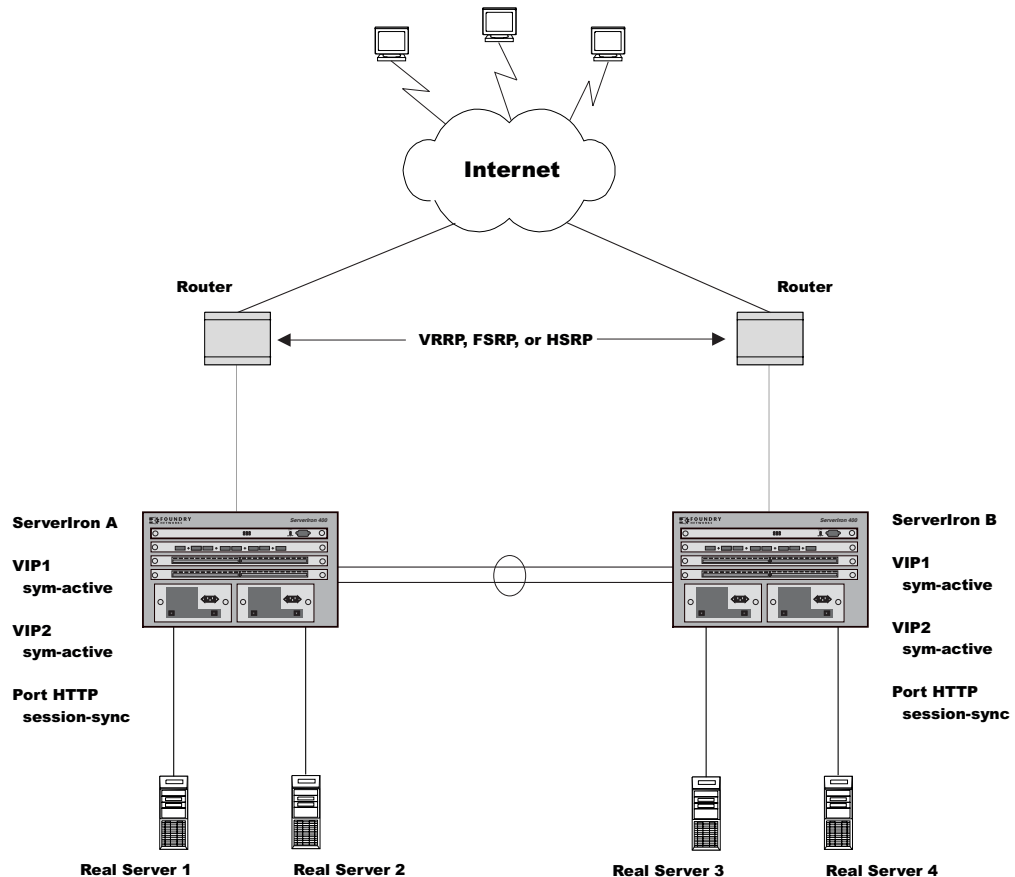
## Active-Active SSLB

Active-active SLB uses session information to ensure that the same ServerIron load balances all requests for a given VIP. The first ServerIron that receives a request for the VIP load balances the request, creates a session table entry for the VIP, and sends the session information to the other ServerIron. Both ServerIrons in the configuration use the session information to use the same ServerIron for subsequent requests for the VIP.

**NOTE:** Active-Active Symmetric SLB is supported only on the ServerIron 400 and ServerIron 800.

Figure 7.5 shows an example of an active-active SSLB configuration.

**Figure 7.5 Active-active SSLB**



In this example, ServerIron A and ServerIron B each have been configured to provide active-active SSLB for the HTTP port on VIP1 and VIP2. The first ServerIron to receive a request for port HTTP on one of these VIPs load balances the request, creates session entries for the VIP, and sends the session information to the other ServerIron. Both ServerIrons use the session information for the VIP to ensure that the same ServerIron load balances subsequent requests for the same application port and VIP.

Either ServerIron can use session information to forward the server reply back to the client. For example, if ServerIron A is the load balancer for a client request and the server reply comes back through ServerIron B, ServerIron B can use the session information received from ServerIron A through session synchronization to perform the required address translations and send the reply to the client. ServerIron B does not need to forward the reply to ServerIron A for address translation and forwarding.

## Configuring Active-Active SSLB

Configuration of active-active SSLB is similar to configuration for standard SSLB. Active-active SSLB requires the following additional steps:

- Specify the active-active synchronization port. This is the port that connects the ServerIron to its SSLB partner. The ServerIrons use the synchronization link to exchange session data.
- Enable session synchronization on the application ports for which you want to use the active-active SLB feature. This is required both to ensure continued service following a failover and to enable each ServerIron to send server replies back to the clients, regardless of which ServerIron load balanced the request.
- On each ServerIron, enable the active-active SLB feature in each of the VIPs for which you want to use the feature. Enable the feature in the same VIPs on each ServerIron.

Active-active SSLB also requires you to specify an SSLB priority on each of the VIPs. Specifying a priority is required to enable the SSLB feature itself. Active-active SSLB disregards the priority value you specify but requires the SSLB feature to be enabled.

### **Configuring an Active-Active Port for SSLB**

When you configure a ServerIron in an active-active configuration, one of the configuration steps is to identify the link between the ServerIrons. To identify the link, you specify the following parameters:

- The local port number (the port on the ServerIron you are configuring that connects it to the other ServerIron)
- The MAC address of the port on the other ServerIron
- The VLAN ID, if the port is a tagged member of multiple VLANs.

---

**NOTE:** The VLAN you specify must be used only for synchronization traffic. Do not specify a VLAN that also will carry data traffic.

---

To configure an active-active port for SSLB, enter a command such as the following at the global CONFIG level of the CLI:

```
ServerIron(config)# server backup-port ethernet 3/5
```

This command configures the active-active link on port 3/5.

To configure a tagged active-active port for SSLB, enter a command such as the following at the global CONFIG level of the CLI:

```
ServerIron(config)# server backup-port ethernet 3/5 200
```

This command configures the active-active link on port 3/5 on VLAN 200 only. The active-active traffic is not forwarded to the other VLANs that port 3/5 is in.

**Syntax:** [no] server backup-port ethernet <portnum> [<vlan-id>]

The <vlan-id> parameter specifies the VLAN you want to use for active-active synchronization traffic.

### **Enabling Session Synchronization on an Application Port**

To enable session synchronization on an application port, edit the port's profile. Here is an example:

```
ServerIron(config)# server port 80
ServerIron(config-port-80)# session-sync
```

This example enables session synchronization for port 80 (HTTP).

**Syntax:** server port <TCP/UDP-portnum>

**Syntax:** [no] session-sync

### **Enabling Active-Active SSLB on a VIP**

To enable active-active SSLB on a VIP, enter the following commands at the VIP configuration level:

```
ServerIronA(config)# server virtual-name VIP1 1.1.1.1
ServerIronA(config-vs-VIP1)# port 80
ServerIronA(config-vs-VIP1)# sym-priority 69
ServerIronA(config-vs-VIP1)# sym-active
```

This example configures VIP1 by adding port 80, enabling SSLB, then enabling active-active SSLB. The **sym-priority** command enables SSLB. The command requires a number from 1 – 255 to enable SSLB. Once you

enter the **sym-active** command to enable active-active SSLB, the software ignores the priority value you specified.

**Syntax:** [no] sym-priority <num>

**Syntax:** [no] sym-active

## Changing the SSLB Discovery Interval

A ServerIron in an SSLB configuration uses SSLB discovery packets to request SSLB information from the other ServerIrons. SSLB discovery packets are proprietary Layer 2 broadcast packets and are sent on all ports in all port-based VLANs.

By default, a ServerIron in an SSLB configuration sends SSLB discovery packets at 200-millisecond intervals. The ServerIron will wait up to 20 equivalent intervals to receive an SSLB discovery packet from another ServerIron. If the ServerIron does not receive an SSLB discovery packet from the other ServerIron within the 20 intervals, the ServerIron concludes that its partner ServerIron is unavailable and assumes control of the VIPs being managed by that ServerIron. For example, if the interval for sending SSLB discovery packets is 200 milliseconds (the default), the ServerIron will wait 20 x 200 milliseconds (four seconds) to receive an SSLB discovery packet from another ServerIron.

You can change the discovery interval multiplier and the wait time multiplier.

- The discovery interval is equal to 200 milliseconds multiplied by the discovery interval multiplier. The default discovery interval multiplier is 1, so the default discovery interval is 200 milliseconds. You can specify a multiplier from 1 – 60.
- The wait time interval is equal to the discovery interval multiplied by the wait time multiplier. The default wait time multiplier is 20. Assuming the discovery interval is 200 milliseconds (the default), the default wait time is four seconds (20 x 200 milliseconds).

---

**NOTE:** The SSLB timer affects the rate at which the ServerIron sends SSLB protocol packets to its SSLB partners. The timer does not affect client or server traffic to or from a VIP.

---

---

**NOTE:** All the ServerIrons in your configuration must use the same SSLB discovery interval and wait time. If you change the interval and wait time on one ServerIron, make the same change on all the other ServerIrons in the SSLB configuration.

---

To change the SSLB discovery interval multiplier and wait time multiplier, enter a command such as the following:

```
ServerIron(config)# server sym-pdu-rate 2 30
```

This command changes the interval at which the ServerIron sends SSLB discovery packets to once every 400 milliseconds, and changes the maximum amount of time the ServerIron will wait for an SSLB discovery packet from another ServerIron to 12 seconds (30 x 400 milliseconds).

**Syntax:** [no] server sym-pdu-rate <disc-mult> <wait-time-mult>

The <disc-mult> parameter specifies the multiplier for the SSLB protocol interval. You can specify a multiplier from 1 – 60. The default is 1.

The <wait-time-mult> parameter specifies how many multiples of the discovery interval the ServerIron will wait for an SSLB discovery packet. You can specify a multiplier from 1 – 60. The default is 20.

## Using Dynamic SSLB Priority

Software release 07.1.08 enhances Symmetric SLB (SSLB) by automatically adjusting a VIP application's SSLB priority to a lower value if a given application fails a health check. With this enhancement, the SSLB priority provides failover for the individual application even if the ServerIron and the application's VIP are both still active.

In previous releases, the priority determines which ServerIron becomes the active one for the VIP and application by default. The priority is static and does not change if the status of the VIP's application changes. As a result, it is possible for SSLB to continue trying to use a real server farm that is no longer responding, instead of failing over to the other ServerIron to load balance requests for the VIP and application.



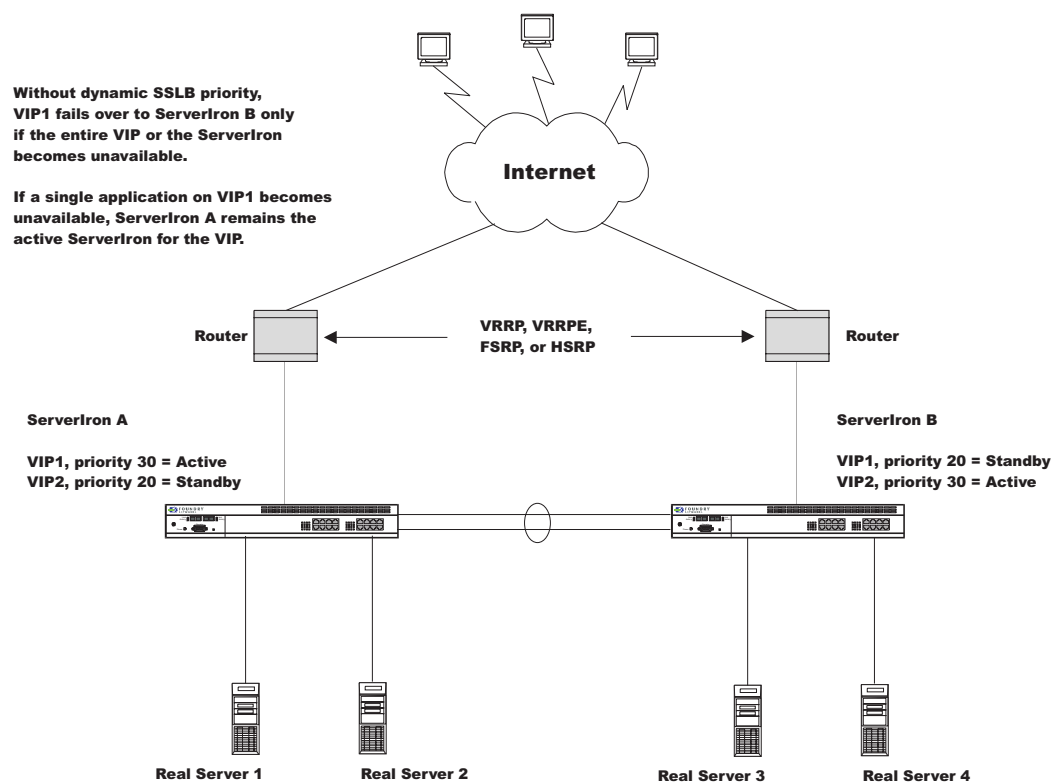
In software release 07.1.08, you can configure a decrement value for the SSLB priority. If an application on a VIP that is enabled for SSLB fails a health check, the ServerIron decrements the VIP's SSLB priority by the amount you specify for the decrement. If the priority value becomes lower than the VIP's priority on the other ServerIron, the software fails the VIP over to the other ServerIron.

**NOTE:** When you configure a decrement value, the value takes effect only if all the application's ports on the real servers fail their health checks. Thus, if the application is still available on at least one of the real servers bound to the VIP, the software does not decrement the priority.

**NOTE:** When you configure the decrement value, do not specify a value that will make the VIP's priority 0. For example, if the VIP's SSLB priority is 10, do not specify 10 as the decrement value. Specify a lower number. Priority value 0 disables SSLB, in which case the VIP becomes active on both ServerIrons.

Figure 7.6 shows an example of an SSLB configuration that uses the default priority handling (not the dynamic priority handling).

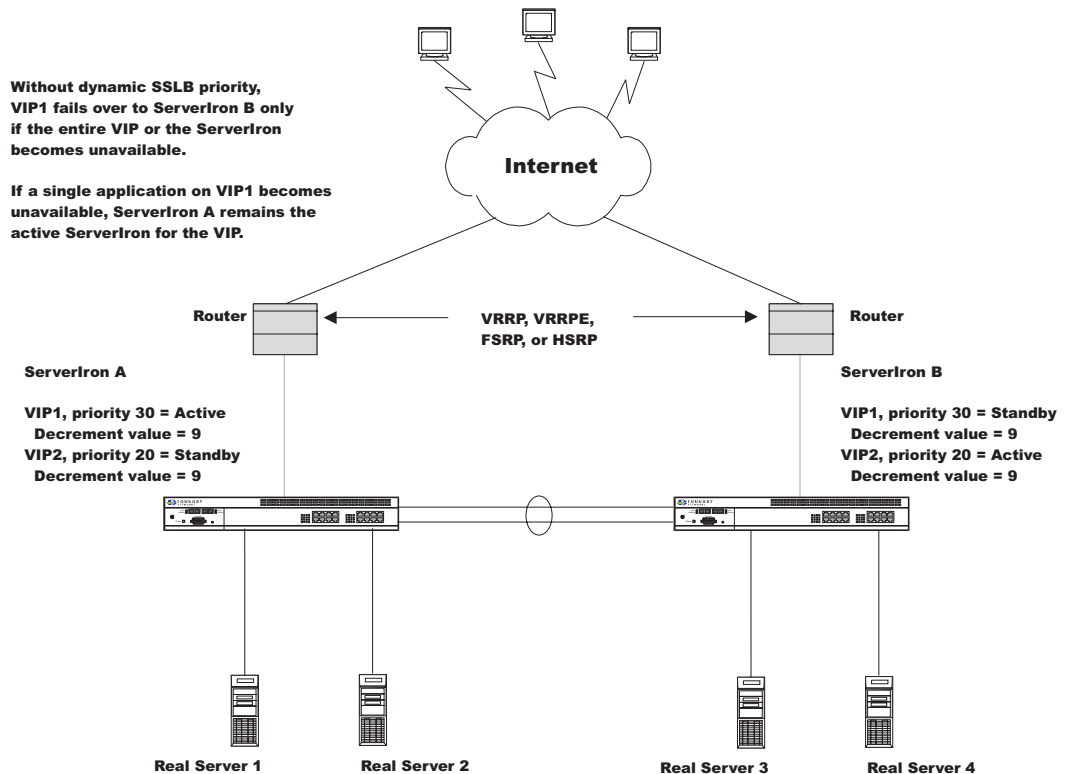
**Figure 7.6 SSLB without dynamic priority**



Using the default priority handling, the software fails over a VIP to the other ServerIron only if the entire VIP or the ServerIron itself becomes unavailable. If an application on the VIP becomes unavailable on all the real servers bound to the VIP, but the VIP itself is still available, the software continues using the same ServerIron for the VIP. As a result, clients are unable to access the unavailable application even if the application is available through the other ServerIron.

Figure 7.7 shows an example of a configuration that uses dynamic SSLB priority.

Figure 7.7 SSLB with dynamic priority



In this configuration, a ServerIron fails over a VIP to the other ServerIron if more than one application on the VIP becomes unavailable. If one application becomes unavailable, the software reduces the VIP's priority by 9 (the decrement value), in this case to 21. At this point, the ServerIron that is active by default for the VIP still has the higher priority, so failover does not occur. However, if a second application becomes unavailable, then the priority becomes 12, which is less than the priority for the VIP on the other ServerIron (20).

When an application becomes available again (and passes a health check), the ServerIron increments the VIP's priority by the decrement amount, thus replacing the priority amount that the software removed when the application failed. If the increment makes the VIP's priority higher than the priority on the other ServerIron, the software fails back over to the ServerIron that originally had the higher priority for the VIP.

If more than one ServerIron has the highest priority for a VIP, the ServerIron that has the highest value for the lowest four bytes of its base MAC address becomes the active ServerIron for the VIP.

**NOTE:** If all the applications that are configured for SSLB on the VIP become unavailable, the software sets the SSLB priority for that VIP to 1 (the lowest value).

### Configuring Dynamic SSLB Priority

The following commands configure the SSLB priority parameters for the configuration in Figure 7.7.

#### Commands on ServerIron A

```
ServerIronA(config-vs-VIP1)# sym-priority 30
ServerIronA(config-vs-VIP1)# dyn-sym-pri-factor 9
```

#### Commands on ServerIron B

```
ServerIronB(config-vs-VIP1)# sym-priority 20
ServerIronB(config-vs-VIP1)# dyn-sym-pri-factor 9
```

The **dyn-sym-pri-factor** commands in this example configure the decrement value to 10. Each time an application on the VIP fails a health check (fails on all the real servers bound to the VIP), the ServerIron

decrements the VIP's SSLB priority by 10. If the priority reaches a value that is lower than the VIP's priority on the other ServerIron, the software fails over active load balancing for the VIP to the other ServerIron. In this example, failover of the VIP from ServerIron A to ServerIron B occurs if **two** applications are unavailable (have failed their health checks).

**Syntax:** [no] dyn-sym-pri-factor <num>

The <num> parameter can be a value from 1 – 255 and specifies the amount by which you want the ServerIron to decrement a VIP's priority when an application on the VIP fails a health check. There is no default. If you specify a value, then the software performs dynamic SSLB priority for the VIP.

---

**NOTE:** Make sure you specify priority and decrement values that provide the level of sensitivity you want. For example, if you want the software to fail over load balancing of a VIP if even a single application fails a health check, then configure the values so that the difference between the two priorities (**sym-priority** command) is less than the decrement value (**dyn-sym-pri-factor** command).

---



---

**NOTE:** Do not specify a value that will make the VIP's priority 0. For example, if the VIP's SSLB priority is 10, do not specify 10 as the decrement value. Specify a lower number. Priority value 0 disables SSLB, in which case the VIP becomes active on both ServerIrons.

---

## Displaying Dynamic SSLB Priority Information

To display the dynamic SSLB configuration and current value for a VIP, enter a command such as the following at any level of the CLI:

```
ServerIronA(config-vs-VIP1)# show server virtual VIP1
Virtual Servers Info

Server Name: VIP1          IP : 2.3.4.5          : 1
Status: enabled Predictor: least-conn TotConn: 0
Dynamic: No HTTP redirect: disabled
Intercept: No
ACL: id = 0
Sym: group = 1 state = 5 priority = 30 keep = 0 dyn priority/factor = 20/ 10
Activates = 1, Inactive= 0
Best-standby-mac = 0000.0000.0000
Port    State    Sticky  Concur  Proxy    CurConn  TotConn  PeakConn
ftp      enabled  NO      NO      NO        0         0         0
http     enabled  NO      NO      NO        0         0         0
default enabled  NO      NO      NO        0         0         0
```

**Syntax:** show server virtual [<name>]

This example shows the configuration and priority information for VIP1 in Figure 7.7. The priority information is shown by the fields in bold type. These fields show the following information.

**Table 7.1: Virtual Server Information for SSLB**

This Field...	Displays...
Sym	<p>Information for Symmetric SLB. The following information is displayed:</p> <ul style="list-style-type: none"> <li>group – The Symmetric SLB group number.</li> <li>state – The state, which should be 5 for the active ServerIron and 3 for other ServerIrons.</li> <li>priority – The Symmetric SLB priority configured on the ServerIron.</li> <li>keep – The number of times an SSLB backup has failed to communicate with the active ServerIron. By default, the counter is incremented by 1 every 400 milliseconds the backup ServerIron is late responding to the active ServerIron's keepalive message. The counter is reset to 0 each time the backup ServerIron replies to a keepalive message. If the counter goes higher than the maximum number allowed (20 by default, thus 8 seconds), the standby ServerIron takes over as the new active ServerIron. Normally, this field almost always contains 0.</li> </ul> <p><b>Note:</b> This field is applicable only on the active ServerIron.</p> <ul style="list-style-type: none"> <li>dyn priority/factor – The current dynamically set priority and the decrement value. In this example, an application has failed a health check, so the dynamic priority is 20 instead of 30. The decrement value is 10. If the priority and dyn priority values match, then all the VIP's applications that are configured for SSLB are responding to their health checks.</li> <li>Activates – The number of times this ServerIron has become the active ServerIron.</li> <li>Inactive – The number of times this ServerIron has changed from being the active ServerIron.</li> <li>Best-standby-mac – The MAC address of the backup ServerIron with the second-highest priority. This ServerIron will become the active ServerIron if a failover occurs.</li> </ul>

## Displaying Symmetric SLB Information

To display Symmetric SLB information, enter the following command at any level of the CLI:

```
ServerIron(config)# show server symmetric
```

```

Server Symmetric port = 0
Group_id = 1 Candidate cnt = 0
Port    No-rx
  1    100824

```

This display shows the following information.

**Table 7.2: Symmetric SLB Information**

This Field...	Displays...
Server Symmetric port	The ServerIron port number on which the ServerIron perceives other ServerIrons running Symmetric SLB.
Group_id	The Symmetric SLB group ID. The group ID is always 1 in the current release.
Candidate cnt	The number of ports on which the ServerIron perceives a partner ServerIron running Symmetric SLB.
Port	The port connected to the other ServerIron.
No-rx	Information Foundry technical support can use to help resolve Symmetric SLB configuration issues.

## Delaying an SSLB ServerIron's Activation Following Recovery

When you enable session synchronization in a ServerIron SSLB configuration, the active ServerIron for a VIP sends session synchronization information to the standby ServerIron. If the VIP's active ServerIron becomes unavailable, the open sessions for the VIP fail over to the other ServerIron, which provides uninterrupted service for the sessions.

The active ServerIron sends session synchronization information to a VIP's standby ServerIron when the session is created. Following a failover, when the standby ServerIron for a VIP has taken over, the standby ServerIron can create new sessions for the VIP. However, because the ServerIron with the higher priority for the VIP is unavailable, the standby ServerIron cannot send synchronization information for the newly created sessions. As a result, when the other ServerIron becomes available again, it resumes service for the VIP but cannot continue the sessions that were created by the standby ServerIron.

You can minimize interruption to sessions created on the standby ServerIron by configuring each ServerIron to delay reactivation following its recovery after a failover. By delaying reactivation of a recovered ServerIron, you provide time for sessions created by the standby ServerIron to terminate normally.

To enable reactivation delay following recovery of a ServerIron, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# server delay-symmetric
```

**Syntax:** [no] server delay-symmetric [<mins>]

The <mins> parameter specifies the number of minutes you want the recovered ServerIron to wait before becoming active again. You can specify from 2 – 120 minutes. The default is 60 minutes.

**NOTE:** You must enter the same command using the same number of minutes on both ServerIrons in the configuration.

## Using SwitchBack

Some ServerIron implementations are in topologies where both directions of load-balancing traffic, the client-to-server and server-to-client traffic, flow through the ServerIron. In this type of configuration, the ServerIron uses two sessions for each connection. One session is for the client-server traffic and the other session is for the server-client traffic.

Typically, the client-server traffic uses less bandwidth than the server-client traffic. The client-server traffic usually consists of the initial GET requests to the VIP and TCP ACKs when the client receives a response from the server. The remaining traffic consists of the requested web pages sent to the client by the server.

The SwitchBack feature places the real server directly in contact with the client, so that server-client traffic does not pass through the ServerIron but instead goes directly from the server to the client. By placing the client directly in contact with the real server, SwitchBack enhances overall performance and throughput, and thus enhances the service experienced by the client.

In addition, for environments with a high volume of simultaneous client-server connections, SwitchBack enhances ServerIron capacity by reducing the number of sessions required for each client connection from two to only one. The ServerIron uses a session for client-server traffic but does not need a session for server-client traffic, because the server communicates directly with the client.

A ServerIron configured for Server Load Balancing acts as a dispatcher, sending client requests for a VIP directly to the real server, which responds directly to the client. The ServerIron does not translate the destination IP address in the client's request from the VIP into the real server's IP address as in other SLB configurations. Instead, the ServerIron leaves the destination IP address unchanged.

---

**NOTE:** You cannot have a router hop between the ServerIrons. They must have Layer 2 connectivity.

---

The SwitchBack feature applies to individual TCP/UDP ports. To configure the ServerIron for SwitchBack, you enable the feature for individual TCP/UDP ports when configuring the virtual server. For example, when you enable TCP port 80 (HTTP) on a virtual server, you can add the `dsr` parameter to enable SwitchBack for that port. Traffic for other ports still returns through the ServerIron. The ServerIron does not translate the destination IP address in client requests for the port with SwitchBack enabled. However, the ServerIron does still translate the destination IP address in the client's request to the real server's IP address for other ports.

---

**NOTE:** "dsr" stands for "Direct Server Return", another name for the SwitchBack feature.

---

To configure the real servers for SwitchBack, configure a loopback interface on each real server and assign the VIP addresses to the loopback interface. The loopback interface enables the real server to respond to client requests directed at the VIPs, while at the same time keeping the real server "hidden". The loopback interface responds to unicast traffic directed to it, but does not respond to ARP requests. The ServerIron responds to pings and ARPs for the VIPs. Thus, an attempt to obtain the real server's MAC address by ARPing a VIP does not succeed. See "Configuring the Loopback Address on a Real Server" on page 7-20.

## Using Remote Failover Servers for SwitchBack

You can use real servers on other sub-nets as remote servers in SwitchBack configurations. In this configuration, the ServerIron uses the local servers as in previous releases. This means all the local servers must have Layer 2 connectivity to the ServerIron. However, if all the local servers become unavailable, the ServerIron then fails over to the remote servers, thus continuing to provide service to the clients.

---

**NOTE:** All the local servers in the SwitchBack configuration still must have Layer 2 connectivity to the ServerIron.

---

## Using Health Checks with SwitchBack

You can use Layer 4 and Layer 7 health checks in your SwitchBack configuration.

- The ServerIron addresses Layer 3 (IP ping) health checks to the real server IP address, and addresses Layer 4 health checks to the real server IP address and the TCP or UDP protocol on the server.
- The ServerIron addresses Layer 7 health checks to the real server MAC address and to the loopback address that matches the VIP address.

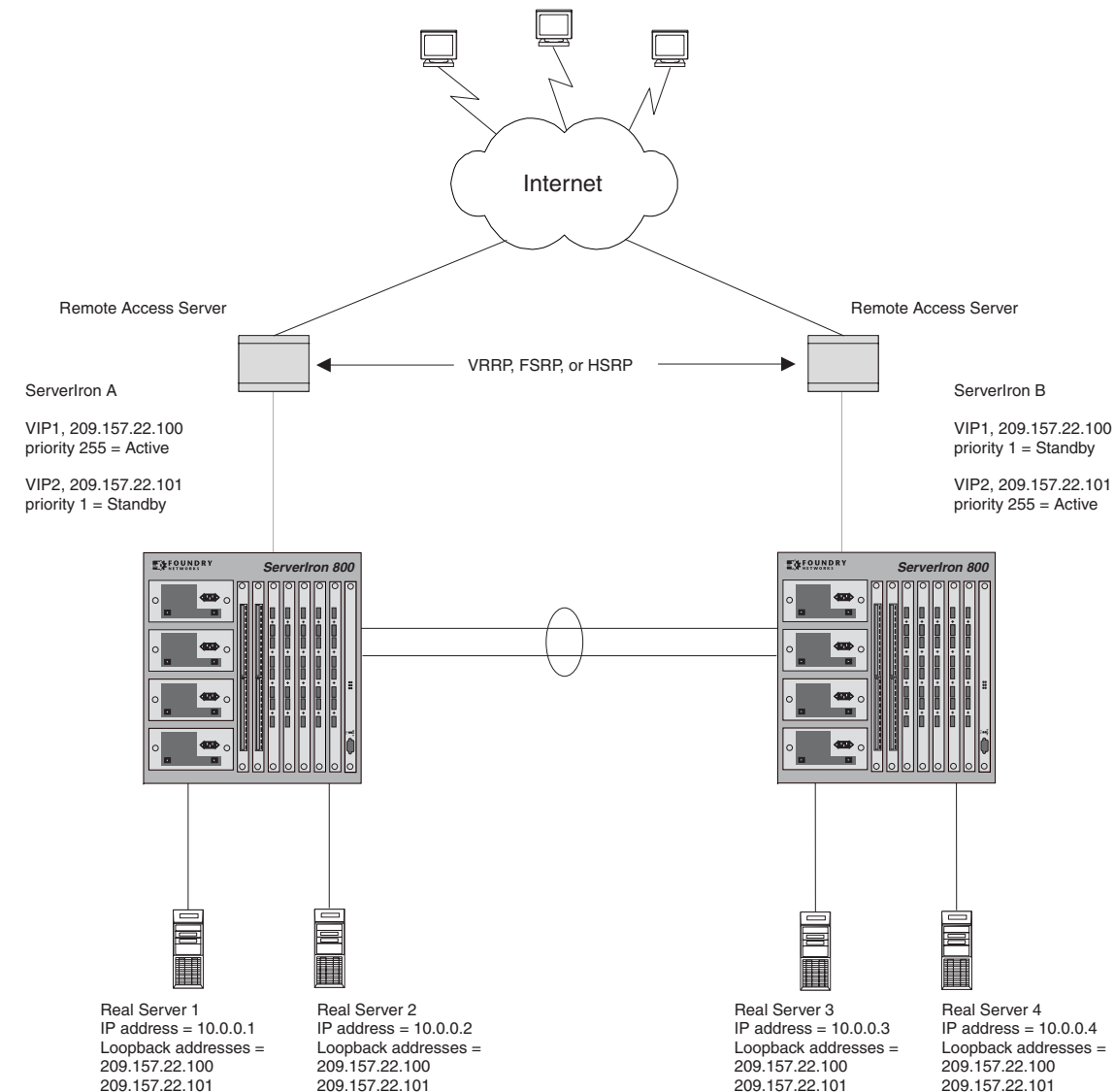
The configuration procedures for the health checks are the same as for other types of SLB. See "Configuring Health Checks" on page 12-1.

## SwitchBack Configuration Example

The following table and Figure 7.8 show an example of a SwitchBack configuration.

Because multiple VIPs are mapping to the same ports on the same real servers, TCP/UDP port binding is used. Thus, port 180 on VIP2 on ServerIron A and on VIP1 on ServerIron B is a logical port that is bound to port 80 on the real servers. For more information, see "Many-To-One TCP/UDP Port Binding" on page 6-98.

ServerIron	Domain Name	Virtual IP (VIP) Address	Priority	VIP's TCP Port	Real IP Address	Real Server's TCP Port
A	www.abc.com	VIP1: 209.157.22.100	254	80	Real Server 1: 10.0.0.1	80
					Real Server 2: 10.0.0.2	80
A	www.def.com	VIP2: 209.157.22.101	2	80	Real Server 1: 10.0.1.1	180
					Real Server 2: 10.0.1.2	180
B	www.abc.com	VIP1: 209.157.22.100	2	80	Real Server 3: 10.0.0.1	180
					Real Server 4: 10.0.0.2	180
B	www.def.com	VIP2: 209.157.22.101	254	80	Real Server 3: 10.0.1.1	80
					Real Server 4: 10.0.1.2	80

**Figure 7.8 ServerIron 800s deployed in SwitchBack configuration**

## CLI Commands

To implement the configuration shown in Figure 7.8, enter the following commands. For simplicity and to illustrate how the SwitchBack feature can be used in conjunction with the Symmetric Server Load Balancing feature, the commands below are the same as the commands in “Using Symmetric Server Load Balancing” on page 7-1. The only difference is the addition of the **dsr** parameter on the **port** commands that add the HTTP port (TCP port 80) to the VIPs.

To enable SwitchBack for additional TCP/UDP ports, you use the **dsr** parameter for each port when you add the port to a VIP.

**NOTE:** Make sure you configure all the real servers on both ServerIrons, and bind the VIPs on each ServerIron to all the real servers.

**NOTE:** Foundry recommends that you specify 2 (instead of 1) as a low priority or 254 (instead of 255) as a high priority. This way, you can easily force failover of the high priority ServerIron to the low priority ServerIron by changing the priority on just one of the ServerIrons. For example, you can force a failover by changing the priority



on the high priority ServerIron from 254 to 1. Since the priority on the low priority ServerIron is 2, the low priority ServerIron takes over for the VIP. Likewise, you can force the low priority ServerIron to take over by changing its priority to 255, since the priority on the high priority ServerIron is only 254.

---

### Commands on ServerIron A

Enter the following commands to configure the real servers. Notice that all four real servers must be configured, and bound to the VIPs, on both ServerIrons. Notice also that two HTTP ports are added to each real server. This type of configuration requires that you use the TCP/UDP port binding feature to bind the ports on the two real servers to the same port on the virtual server. For information, see “Many-To-One TCP/UDP Port Binding” on page 6-98.

```
ServerIronA(config)# server real-name Real_Server_1 10.0.0.1
ServerIronA(config-rs-Real_Server_1)# port http
ServerIronA(config-rs-Real_Server_1)# port 180
ServerIronA(config-rs-Real_Server_1)# exit
ServerIronA(config)# server real-name Real_Server_2 10.0.0.2
ServerIronA(config-rs-Real_Server_2)# port http
ServerIronA(config-rs-Real_Server_2)# port 180
ServerIronA(config-rs-Real_Server_2)# exit
ServerIronA(config)# server real-name Real_Server_3 10.0.1.1
ServerIronA(config-rs-Real_Server_3)# port http
ServerIronA(config-rs-Real_Server_3)# port 180
ServerIronA(config-rs-Real_Server_3)# exit
ServerIronA(config)# server real-name Real_Server_4 10.0.1.2
ServerIronA(config-rs-Real_Server_4)# port http
ServerIronA(config-rs-Real_Server_4)# port 180
ServerIronA(config-rs-Real_Server_4)# exit
```

Enter the following commands to configure the VIPs.

```
ServerIronA(config)# server virtual-name VIP1 209.157.22.100
ServerIronA(config-vs-VIP1)# port http dsr
ServerIronA(config-vs-VIP1)# bind http Real_Server_1 http Real_Server_2 http
Real_Server_3 http Real_Server_4 http
ServerIronA(config-vs-VIP1)# sym-priority 254
ServerIronA(config-vs-VIP1)# exit
ServerIronA(config)# server virtual-name VIP2 209.157.22.101
ServerIronA(config-vs-VIP2)# port http dsr
ServerIronA(config-vs-VIP2)# bind http Real_Server_1 180 Real_Server_2 180
Real_Server_3 180 Real_Server_4 180
ServerIronA(config-vs-VIP2)# no http port translate
ServerIronA(config-vs-VIP2)# sym-priority 2
ServerIronA(config-vs-VIP2)# exit
ServerIronA(config)# write memory
```

### Commands on ServerIron B

Enter the following commands to configure the real servers.

```
ServerIronB(config)# server real-name Real_Server_1 10.0.0.1
ServerIronB(config-rs-Real_Server_1)# port http
ServerIronB(config-rs-Real_Server_1)# port 180
ServerIronB(config-rs-Real_Server_1)# exit
ServerIronB(config)# server real-name Real_Server_2 10.0.0.2
ServerIronB(config-rs-Real_Server_2)# port http
ServerIronB(config-rs-Real_Server_2)# port 180
ServerIronB(config-rs-Real_Server_2)# exit
ServerIronB(config)# server real-name Real_Server_3 10.0.1.1
ServerIronB(config-rs-Real_Server_3)# port http
```

```
ServerIronB(config-rs-Real_Server_3)# port 180
ServerIronB(config-rs-Real_Server_3)# exit
ServerIronB(config)# server real-name Real_Server_4 10.0.1.2
ServerIronB(config-rs-Real_Server_4)# port http
ServerIronB(config-rs-Real_Server_4)# port 180
ServerIronB(config-rs-Real_Server_4)# exit
```

Enter the following commands to configure the VIPs.

```
ServerIronB(config)# server virtual-name VIP1 209.157.22.100
ServerIronB(config-vs-VIP1)# port http dsr
ServerIronB(config-vs-VIP1)# bind http Real_Server_1 180 Real_Server_2 180
Real_Server_3 180 Real_Server_4 180
ServerIronB(config-vs-VIP1)# no http port translate
ServerIronB(config-vs-VIP1)# sym-priority 2
ServerIronB(config-vs-VIP1)# exit
ServerIronB(config)# server virtual-name VIP2 209.157.22.101
ServerIronB(config-vs-VIP2)# port http dsr
ServerIronB(config-vs-VIP2)# bind http Real_Server_1 http Real_Server_2 http
Real_Server_3 http Real_Server_4 http
ServerIronB(config-vs-VIP2)# sym-priority 254
ServerIronB(config-vs-VIP2)# exit
ServerIronB(config)# write memory
```

## Configuring the Loopback Address on a Real Server

This section contains procedures for configuring loopback addresses on some common types of real servers.

---

**NOTE:** The information in this section is based on information from the vendors of these servers. For more information, please consult your real server vendor.

---

### Solaris

To configure a loopback address on Solaris, enter the following command:

**ifconfig lo0:1 <vip-addr> netmask <net-mask> up**

You might need to “plumb” the interface first. In this case, enter the following commands:

**ifconfig lo0:1 plumb**

**ifconfig lo0:1 <vip-addr> netmask <net-mask> up**

---

**NOTE:** This command applies to the current running configuration only. To make the address permanent so that it is reconfigured following a reboot or power cycle, create the file /etc/hostname.lo0:1.

---



---

**NOTE:** For Hewlett-Packard (HP) version 11.x, use the May 2000 or later patch.

---

### Linux

To configure a loopback interface on Linux, enter a command such as the following:

**ifconfig lo:0 <vip-addr> netmask <net-mask> up**

---

**NOTE:** This command applies to the current running configuration only. To make the address permanent so that it is reconfigured following a reboot or power cycle, go to /etc/sysconfig/network-scripts and make a file called ifcfg-lo:0 using ifcfg-lo as a template.

---

## NT

To configure a loopback interface on NT, you need to configure a new network adapter. Use the following procedure. This procedure applies to the following products:

- Microsoft Windows 2000 Professional
- Microsoft Windows 2000 Server
- Microsoft Windows 2000 Advanced Server
- Microsoft Windows 2000 Datacenter Server

---

**NOTE:** When you add a loopback interface to NT, NT sometimes creates a route that has the same address as the loopback interface. You need to delete this route. In some cases, the procedure for deleting the route can include deleting the correct route to the server's default gateway. When this is the case, you need to add this route back to NT.

---

### *Manual Installation*

1. Click Start, point to Settings, click Control Panel, and then double-click Add/Remove Hardware.
2. Click Add/Troubleshoot a device, and then click Next.
3. Click Add a new device, and then click Next.
4. Click No, I want to select the hardware from a list, and then click Next.
5. Click Network adapters, and then click Next.
6. In the Manufacturers box, click Microsoft.
7. In the Network Adapter box, click Microsoft Loopback Adapter, and then click Next.
8. Click Finish.

After the adapter is installed successfully, you can configure its options manually, as with any other adapter.

---

**NOTE:** If the TCP/IP properties are configured to use DHCP (the default), the adapter will eventually use an autonet address (169.254.x.x/16) because it is not actually connected to any physical media.

---

### *Unattended Installation*

Modify the Unattend.txt file using the following example as a guide to install the Microsoft Loopback adapter:

```
[NetAdapters]
Adapter01=Params.Adapter01

[Params.Adapter01]
InfID="*msloop" ; Microsoft Loopback Adapter
ConnectionName = "MS Loopback Adapter"

[NetProtocols]
MS_TCPIP=Params.MS_TCPIP

; TCP/IP parameters
; Use parameter values specific to your network
[Params.MS_TCPIP]
AdapterSections=params.TCPIP.Adapter01
DNS=yes
DNSSuffixSearchOrder=mycorp.com
EnableLMHosts=No

; Adapter Specific TCP/IP parameters
; Use parameter values specific to your network
[params.TCPIP.Adapter01]
SpecificTo=Adapter01
DNSDomain=mycorp.com
```

```
DNSServerSearchOrder=192.168.5.251
WINS=no
DHCP=no
IPAddress=192.168.5.10
SubnetMask=255.255.255.0
DefaultGateway=192.168.5.254
```

### ***Deleting the Unwanted Routes***

In some cases, NT creates a route that has the same address as the loopback interface. You need to delete this route.

Two methods are shown in this section. If you receive an error message while trying to use the simple method, you need to use the long method instead.

---

**NOTE:** Regardless of the method you use, you must repeat the procedure every time the NT server is booted. However, you can create a small batch file to enter these commands and add the batch file to the AT subsystem so that the file runs automatically each time the server is booted.

---

### ***Simple Method***

The simple method requires you to delete the route that NT creates when you add the loopback interface. The route you need to delete is the one that has the same IP address as the loopback interface.

1. Enter the **route print** command to display the server's route table. In this example, the loopback interface has address 192.168.200.106.

```
C:\>route print
```

Active Routes:

Network Address	Netmask	Gateway Address	Interface	Metric
0.0.0.0	0.0.0.0	192.168.204.254	192.168.200.251	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
<b>192.168.200.0</b>	<b>255.255.255.0</b>	<b>192.168.200.106</b>	<b>192.168.200.106</b>	<b>1</b>
192.168.200.0	255.255.255.0	192.168.200.251	192.168.200.251	1
192.168.200.106	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.200.251	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.200.255	255.255.255.255	192.168.200.251	192.168.200.251	1
224.0.0.0	224.0.0.0	192.168.200.106	192.168.200.106	1
224.0.0.0	224.0.0.0	192.168.200.251	192.168.200.251	1
255.255.255.255	255.255.255.255	192.168.200.251	192.168.200.251	1

2. Delete the route that has the same address as the loopback interface.

```
C:\>route delete 192.168.200.0 mask 255.255.255.0 192.168.200.106
```

3. Display the route table again to verify that the unwanted route is gone.

```
C:\>route print
```

Active Routes:

Network Address	Netmask	Gateway Address	Interface	Metric
0.0.0.0	0.0.0.0	192.168.204.254	192.168.200.251	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.200.0	255.255.255.0	192.168.200.251	192.168.200.251	1
192.168.200.106	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.200.251	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.200.255	255.255.255.255	192.168.200.251	192.168.200.251	1
224.0.0.0	224.0.0.0	192.168.200.106	192.168.200.106	1

```

224.0.0.0          224.0.0.0  192.168.200.251  192.168.200.251      1
255.255.255.255   255.255.255.255  192.168.200.251  192.168.200.251      1

```

### Long Method

The long method, like the short method, requires you to delete the route that NT creates when you add the loopback interface. However, what makes this method long is that in some cases, when the route table has more than one route in the network that contains the loopback interface, the **route delete** command deletes the wrong route. In this case, you need to enter the command again to delete the route that has the loopback address, then re-add the other route.

1. Enter the **route print** command to display the server's route table. In this example, the loopback interface has address 192.168.200.106. Notice that the route table also contains another route (192.168.200.250) in the same network. The 192.168.200.250 route is the gateway route and needs to stay in the route table.

```
C:\users\default>route print
```

Active Routes:

Network Address	Netmask	Gateway Address	Interface	Metric
0.0.0.0	0.0.0.0	192.168.200.254	192.168.200.250	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
<b>192.168.200.0</b>	<b>255.255.255.0</b>	<b>192.168.200.250</b>	<b>192.168.200.250</b>	<b>1</b>
<b>192.168.200.0</b>	<b>255.255.255.0</b>	<b>192.168.200.106</b>	<b>192.168.200.106</b>	<b>1</b>
192.168.200.106	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.200.250	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.200.255	255.255.255.255	192.168.200.106	192.168.200.106	1
224.0.0.0	224.0.0.0	192.168.200.250	192.168.200.250	1
224.0.0.0	224.0.0.0	192.168.200.106	192.168.200.106	1
255.255.255.255	255.255.255.255	192.168.200.106	192.168.200.106	1

2. Enter the **route delete** command to delete the unwanted 192.168.200.106 route.

```
C:\users\default>route delete 192.168.200.0 mask 255.255.255.0 192.168.200.106
```

3. Display the route table again to verify the results. In this example, NT deletes the first 192.168.200.x route in the table instead of deleting the route you want to delete. If this occurs when you are performing this procedure, go to Step 4. Otherwise, you are through with this procedure.

```
C:\users\default>route print
```

Active Routes:

Network Address	Netmask	Gateway Address	Interface	Metric
0.0.0.0	0.0.0.0	192.168.200.254	192.168.200.250	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
<b>192.168.200.0</b>	<b>255.255.255.0</b>	<b>192.168.200.106</b>	<b>192.168.200.106</b>	<b>1</b>
192.168.200.106	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.200.250	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.200.255	255.255.255.255	192.168.200.106	192.168.200.106	1
224.0.0.0	224.0.0.0	192.168.200.250	192.168.200.250	1
224.0.0.0	224.0.0.0	192.168.200.106	192.168.200.106	1
255.255.255.255	255.255.255.255	192.168.200.106	192.168.200.106	1

4. Enter the **route delete** command again to delete the unwanted route.

```
C:\users\default>route delete 192.168.200.0 mask 255.255.255.0 192.168.200.106
```

5. Display the route table again to verify the results. In this example, none of the 192.168.200.x routes remain in the table.

```
C:\users\default>route print
```

Active Routes:

Network Address	Netmask	Gateway Address	Interface	Metric
0.0.0.0	0.0.0.0	192.168.200.254	192.168.200.250	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.200.106	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.200.250	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.200.255	255.255.255.255	192.168.200.106	192.168.200.106	1
224.0.0.0	224.0.0.0	192.168.200.250	192.168.200.250	1
224.0.0.0	224.0.0.0	192.168.200.106	192.168.200.106	1
255.255.255.255	255.255.255.255	192.168.200.106	192.168.200.106	1

6. Enter the **route add** command to re-add the gateway route.

```
C:\users\default>route add 192.168.200.0 mask 255.255.255.0 192.168.200.250
```

7. Display the route table again to verify that the table contains the gateway route but does not contain a route with the loopback address.

```
C:\users\default>route print
```

Active Routes:

Network Address	Netmask	Gateway Address	Interface	Metric
0.0.0.0	0.0.0.0	192.168.200.254	192.168.200.250	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
<b>192.168.200.0</b>	<b>255.255.255.0</b>	<b>192.168.200.250</b>	<b>192.168.200.250</b>	<b>1</b>
192.168.200.106	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.200.250	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.200.255	255.255.255.255	192.168.200.106	192.168.200.106	1
224.0.0.0	224.0.0.0	192.168.200.250	192.168.200.250	1
224.0.0.0	224.0.0.0	192.168.200.106	192.168.200.106	1
255.255.255.255	255.255.255.255	192.168.200.106	192.168.200.106	1

---

## Chapter 8

# Configuring Transparent VIPs and Stateless SLB

This chapter describes Server Load Balancing configuration options that are “stateless”. Stateless SLB does not use session table entries for the TCP and UDP sessions between the ServerIron and clients or real servers.

These configuration options are especially useful if you want to deploy multiple ServerIrons to provide service for the same VIPs or applications but the network topology cannot ensure that server responses will pass back through the ServerIron.

---

**NOTE:** The SwitchBack feature allows you to deploy a single ServerIron in a network where the server responses do not pass back through the ServerIron. Compare the configuration example for SwitchBack with the examples in this chapter to determine which type of configuration is applicable to your network. See “Configuring Symmetric SLB and SwitchBack” on page 7-1.

---

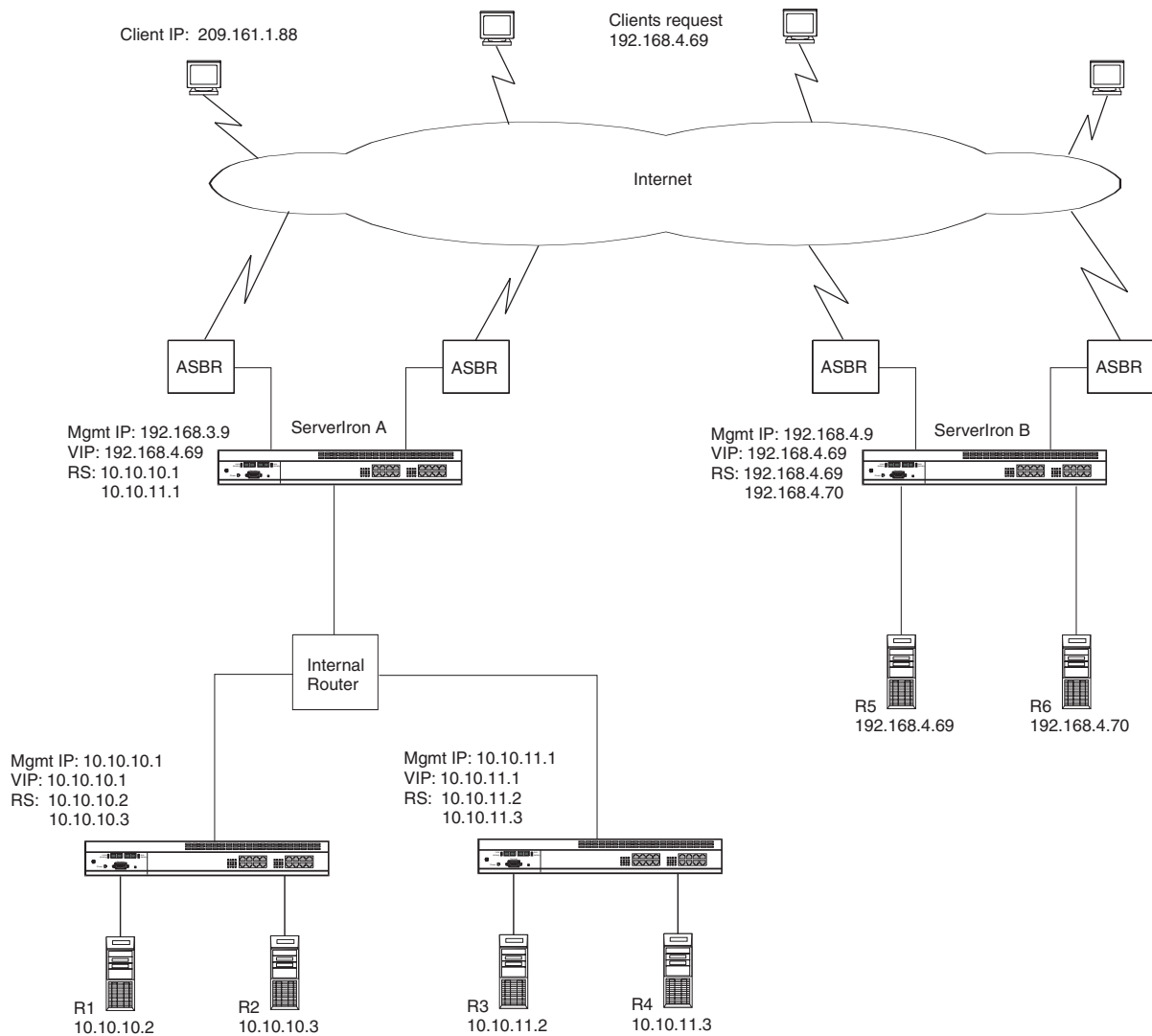
### Transparent VIP

Transparent VIP allows you to configure a ServerIron to transparently load balance a VIP, without owning the VIP address.

Normally, when you configure a VIP on a ServerIron, that ServerIron owns the VIP address. As a result, you cannot configure the same VIP address on another ServerIron without causing an address conflict. However, if you configure the VIP to be transparent, the ServerIron performs load balancing for the VIP but does not own the VIP.

Transparent VIP is useful when you want to configure multiple ServerIrons to load balance the same VIP. For example, if you have globally distributed clients that access the same VIP, you can place ServerIrons close to those clients for optimal service, and use the ServerIron to load balance requests for the VIP to locally distributed server farms. Figure 8.1 shows an example.

**Figure 8.1 Example Transparent VIP Configuration**



In this example, two ServerIrons are configured with transparent VIP 192.168.4.69. ServerIron A transparently load balances requests for the VIP among two other ServerIrons, each of which load balances for identically configured server farms containing content for the VIP.

ServerIron B transparently load balances requests for the VIP among two real servers attached directly to ServerIron B. On ServerIron A and ServerIron B, the transparent VIP is bound to two real servers. On ServerIron A, the real servers are actually VIPs configured on the downstream ServerIrons. On ServerIron B, the real servers are real servers. In either case, the configuration on the upstream ServerIrons is the same.

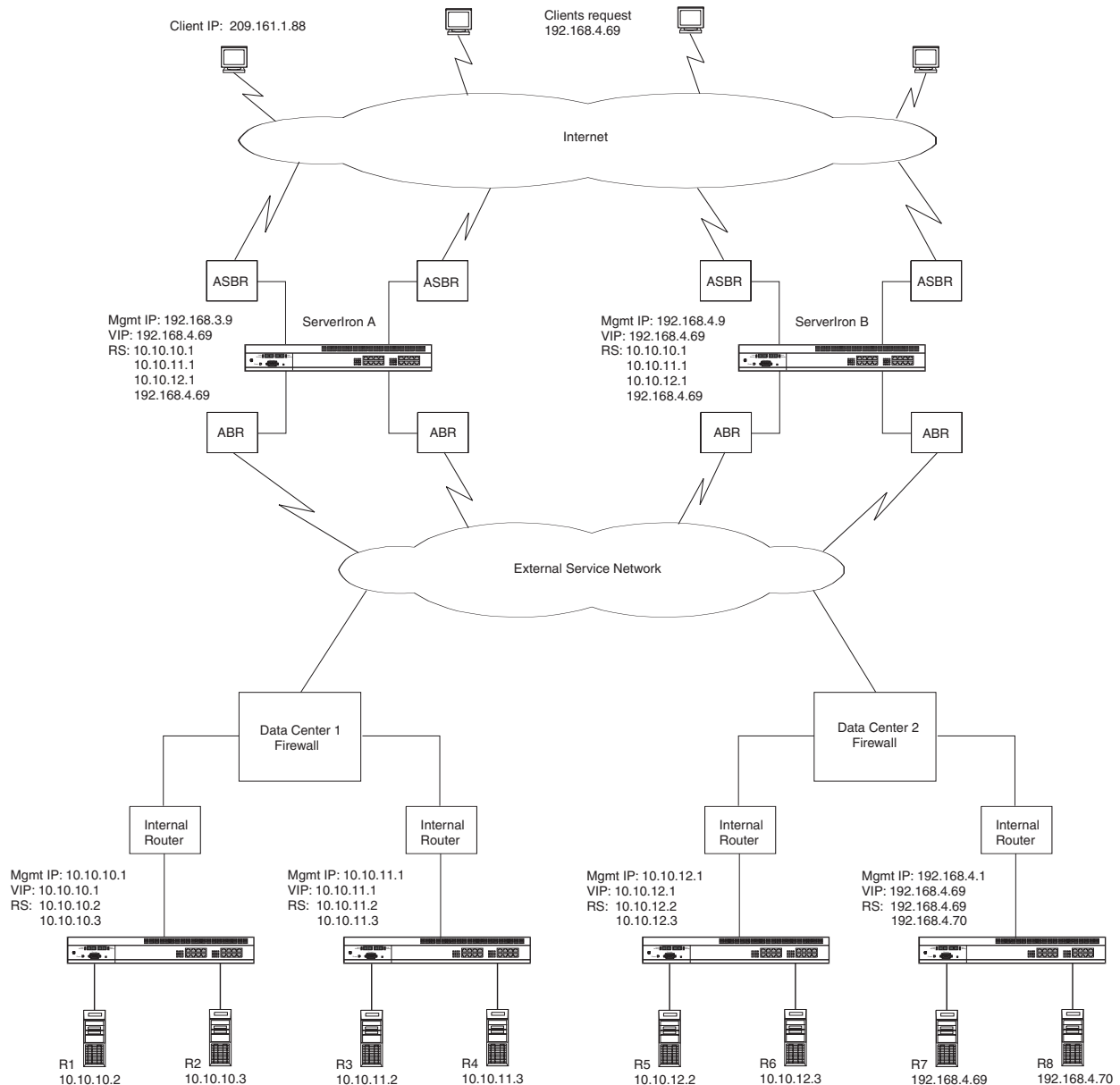
Notice that one of the real servers attached to ServerIron B has the same IP address as the transparent VIP. This real server is the actual owner of the IP address. When you configure a transparent VIP, make sure one of the devices downstream from the ServerIrons has the IP address of the transparent VIP.

**NOTE:** Since the ServerIrons do not own a transparent VIP, they do not respond to pings for the VIP. Instead, the device that is the actual owner of the address responds to pings for the address.



Figure 8.2 shows another example of a transparent VIP configuration. In this example, there are more Serverlrons transparently load balancing requests for the VIP, and the clients are separated from the VIP content by an additional networking layer.

**Figure 8.2 Another Example Transparent VIP Configuration**



The configurations in Figure 8.1 and Figure 8.2 differ in an important way: the server responses in Figure 8.1 are guaranteed to pass back through the ServerIron that transparently load balanced the request. However, the server responses in Figure 8.2 can take different paths depending on network conditions, and thus are not guaranteed to pass back through the ServerIron that transparently load balanced the request.

Depending on the network topology, you might need to configure the transparent VIP's application ports for stateless SLB.

## Stateful and Stateless Load Balancing

By default, SLB uses stateful load balancing for VIPs. This is true for a standard VIP (a VIP “owned” by the ServerIron) and for a transparent VIP. Stateful load balancing means the ServerIron creates session table entries for the connections between the client and the destination (real server or another VIP).

You can configure individual TCP or UDP applications for stateless load balancing, instead of stateful load balancing.

- Stateful load balancing uses session table entries to track connections between the client and server, and requires the server responses to pass back through the ServerIron.
- Stateless load balancing does not create session table entries and does not require the server response to pass back through the ServerIron.

If the network topology of your transparent VIP configuration ensures that a server response always flows back through the ServerIron that forwarded the request, then you can use stateful load balancing. However, if the server response can take different paths depending on network conditions, and thus is not guaranteed to always pass back through the same ServerIron, you need to configure the applications served by the transparent VIP for stateless load balancing.

The load balancing methods you can use differ depending on whether the application ports use stateful SLB or stateless SLB:

- Stateful SLB can use the standard load balancing methods: round-robin, least connections, weighted, and server response time.
- Stateless SLB does not use the standard SLB load balancing methods, but instead uses a hashing algorithm. For information, see “Stateless TCP/UDP Ports” on page 8-8.

## Enabling the Transparent VIP Feature

To enable transparent VIP, enable the feature globally, then configure a cache redirection policy and apply it locally to the ServerIron port(s) connected to the clients. The cache redirection policy identifies the application port(s) on the VIP that you want to load balance.

For example, enter commands such as the following to enable transparent VIP for TCP port 80 (HTTP):

```
ServerIron(config)# server transparent-vip
ServerIron(config)# ip policy 1 cache tcp 80 local
ServerIron(config)# interface ethernet 1
ServerIron(config-if-1)# ip-policy 1
```

These commands enable the feature globally, then use a cache redirection policy to locally enable the feature on the port connected to the clients, in this case port 1.

**Syntax:** [no] server transparent-vip

**Syntax:** [no] ip policy <num> cache <tcp/udp-portnum> local

**Syntax:** [no] ip-policy <num>

## Configuring an Individual Virtual Server To Be Transparent

After you enable the ServerIron for transparent VIP, you still must enable the individual VIP for the feature. Transparent VIP applies only to the VIPs on which you enable it.

To configure an individual virtual server for the transparent VIP feature, enter a command such as the following:

```
ServerIron(config-vs-TransVIP)# transparent-vip
```

**Syntax:** [no] transparent-vip

## Complete CLI Example for Figure 8.1 on page 8-2

The following sections list the CLI commands for configuring the ServerIrons in Figure 8.1 on page 8-2.

Notice that ServerIron A load balances requests for the VIP 192.168.4.69 to other ServerIrons. The other ServerIrons are configured to load balance the request to real servers. The VIP 192.168.4.69 on ServerIron A is configured as a transparent VIP bound to VIPs configured on ServerIrons 10.10.10.1 and 10.10.11.1. Thus, ServerIron A load balances the request for VIP 192.168.4.69 to another ServerIron, and the ServerIron that receives the request load balances the request to a real server. VIP 192.168.4.69 on ServerIron B is not transparent, since real server R7 actually owns the IP address of the VIP.

### ServerIron 192.168.3.9

```
ServerIron> enable
ServerIron# configure terminal
ServerIron(config)# hostname ServerIronA
ServerIronA(config)# ip address 192.168.3.9 255.255.255.0
ServerIronA(config)# ip default-gateway 192.168.3.1
ServerIronA(config)# server transparent-vip
ServerIronA(config)# ip policy 1 cache tcp 80 local
ServerIronA(config)# interface ethernet 1
ServerIronA(config-if-1)# ip-policy 1
ServerIronA(config-if-1)# exit
ServerIronA(config)# interface ethernet 2
ServerIronA(config-if-2)# ip-policy 1
ServerIronA(config-if-2)# exit
ServerIronA(config)# server real ServerIron10 10.10.10.1
ServerIronA(config-rs-ServerIron10)# port http
ServerIronA(config-rs-ServerIron10)# exit
ServerIronA(config)# server real ServerIron11 10.10.11.1
ServerIronA(config-rs-ServerIron11)# port http
ServerIronA(config-rs-ServerIron11)# exit
ServerIronA(config)# server virtual ExternalSite 192.168.4.69
ServerIronA(config-vs-ExternalSite)# transparent-vip
ServerIronA(config-vs-ExternalSite)# bind http ServerIron10 http
ServerIronA(config-vs-ExternalSite)# bind http ServerIron11 http
```

### ServerIron 192.168.4.9

```
ServerIron> enable
ServerIron# configure terminal
ServerIron(config)# hostname ServerIronB
ServerIronB(config)# ip address 192.168.4.9 255.255.255.0
ServerIronB(config)# ip default-gateway 192.168.4.1
ServerIronB(config)# server real R5 192.168.4.69
ServerIronB(config-rs-R5)# port http
ServerIronB(config-rs-R5)# exit
ServerIronB(config)# server real R6 192.168.4.70
ServerIronB(config-rs-R6)# port http
ServerIronB(config-rs-R6)# exit
ServerIronB(config)# server virtual ExternalSite 192.168.4.69
ServerIronB(config-vs-ExternalSite)# transparent-vip
ServerIronB(config-vs-ExternalSite)# bind http R5 http
ServerIronB(config-vs-ExternalSite)# bind http R6 http
```

### ServerIron 10.10.10.1

```
ServerIron> enable
ServerIron# configure terminal
ServerIron(config)# hostname ServerIron10
ServerIron10(config)# ip address 10.10.10.1 255.255.255.0
ServerIron10(config)# ip default-gateway 10.10.10.86
ServerIron10(config)# server real R1 10.10.10.2
ServerIron10(config-rs-R1)# port http
ServerIron10(config-rs-R1)# exit
```

```
ServerIron10(config)# server real R2 10.10.10.3
ServerIron10(config-rs-R2)# port http
ServerIron10(config-rs-R2)# exit
ServerIron10(config)# server virtual InternalSite10 10.10.10.1
ServerIron10(config-vs-InternalSite10)# bind http R1 http
ServerIron10(config-vs-InternalSite10)# bind http R2 http
```

### ServerIron 10.10.11.1

```
ServerIron> enable
ServerIron# configure terminal
ServerIron(config)# hostname ServerIron11
ServerIron11(config)# ip address 10.10.11.1 255.255.255.0
ServerIron11(config)# ip default-gateway 10.10.11.86
ServerIron11(config)# server real R3 10.10.11.2
ServerIron11(config-rs-R3)# port http
ServerIron11(config-rs-R3)# exit
ServerIron11(config)# server real R4 10.10.11.3
ServerIron11(config-rs-R4)# port http
ServerIron11(config-rs-R4)# exit
ServerIron11(config)# server virtual InternalSite11 10.10.11.1
ServerIron11(config-vs-InternalSite11)# bind http R3 http
ServerIron11(config-vs-InternalSite11)# bind http R4 http
```

## Complete CLI Example for Figure 8.2 on page 8-3

The following sections list the CLI commands for configuring the ServerIrons in Figure 8.2 on page 8-3.

Notice that the application port (HTTP) on the transparent VIP is configured to be stateless on ServerIron A and ServerIron B. Since the application is stateless, these ServerIrons do not create session table entries for the application. Server responses to requests for the application can use any path to return to the client. A server response does not need to pass back through the ServerIron that forwarded the request.

The ServerIrons that are directly connected to the real servers are not configured to make the application port stateless. Since a server response is guaranteed to pass back through the ServerIron at this level, these ServerIrons can create session table entries for the application and thus use features that depend on the session table, including the standard SLB load balancing methods.

### ServerIron 192.168.3.9

```
ServerIron> enable
ServerIron# configure terminal
ServerIron(config)# hostname ServerIronA
ServerIronA(config)# ip address 192.168.3.9 255.255.255.0
ServerIronA(config)# ip default-gateway 192.168.3.1
ServerIronA(config)# server transparent-vip
ServerIronA(config)# ip policy 1 cache tcp 80 local
ServerIronA(config)# interface ethernet 1
ServerIronA(config-if-1)# ip-policy 1
ServerIronA(config-if-1)# exit
ServerIronA(config)# interface ethernet 2
ServerIronA(config-if-2)# ip-policy 1
ServerIronA(config-if-2)# exit
ServerIronA(config)# server real ServerIron10 10.10.10.1
ServerIronA(config-rs-ServerIron10)# port http
ServerIronA(config-rs-ServerIron10)# exit
ServerIronA(config)# server real ServerIron11 10.10.11.1
ServerIronA(config-rs-ServerIron11)# port http
ServerIronA(config-rs-ServerIron11)# exit
ServerIronA(config)# server real ServerIron12 10.10.12.1
ServerIronA(config-rs-ServerIron12)# port http
```

```

ServerIronA(config-rs-ServerIron12)# exit
ServerIronA(config)# server real ServerIron69 192.168.4.1
ServerIronA(config-rs-ServerIron69)# port http
ServerIronA(config-rs-ServerIron69)# exit
ServerIronA(config)# server virtual ExternalSite 192.168.4.69
ServerIronA(config-vs-ExternalSite)# port http stateless
ServerIronA(config-vs-ExternalSite)# bind http ServerIron10 http
ServerIronA(config-vs-ExternalSite)# bind http ServerIron11 http
ServerIronA(config-vs-ExternalSite)# bind http ServerIron12 http
ServerIronA(config-vs-ExternalSite)# bind http ServerIron69 http

```

### ServerIron 192.168.4.9

```

ServerIron> enable
ServerIron# configure terminal
ServerIron(config)# hostname ServerIronB
ServerIronB(config)# ip address 192.168.4.9 255.255.255.0
ServerIronB(config)# ip default-gateway 192.168.4.1
ServerIronB(config)# server transparent-vip
ServerIronB(config)# ip policy 1 cache tcp 80 local
ServerIronB(config)# interface ethernet 1
ServerIronB(config-if-1)# ip-policy 1
ServerIronB(config-if-1)# exit
ServerIronB(config)# interface ethernet 2
ServerIronB(config-if-2)# ip-policy 1
ServerIronB(config-if-2)# exit
ServerIronB(config)# server real ServerIron10 10.10.10.1
ServerIronB(config-rs-ServerIron10)# port http
ServerIronB(config-rs-ServerIron10)# exit
ServerIronB(config)# server real ServerIron11 10.10.11.1
ServerIronB(config-rs-ServerIron11)# port http
ServerIronB(config-rs-ServerIron11)# exit
ServerIronB(config)# server real ServerIron12 10.10.12.1
ServerIronB(config-rs-ServerIron12)# port http
ServerIronB(config-rs-ServerIron12)# exit
ServerIronB(config)# server real ServerIron69 192.168.4.1
ServerIronB(config-rs-ServerIron69)# port http
ServerIronB(config-rs-ServerIron69)# exit
ServerIronB(config)# server virtual ExternalSite 192.168.4.69
ServerIronB(config-vs-ExternalSite)# transparent-vip
ServerIronB(config-vs-ExternalSite)# port http stateless
ServerIronB(config-vs-ExternalSite)# bind http ServerIron10 http
ServerIronB(config-vs-ExternalSite)# bind http ServerIron11 http
ServerIronB(config-vs-ExternalSite)# bind http ServerIron12 http
ServerIronB(config-vs-ExternalSite)# bind http ServerIron69 http

```

### ServerIron 10.10.10.1

```

ServerIron> enable
ServerIron# configure terminal
ServerIron(config)# hostname ServerIron10
ServerIron10(config)# ip address 10.10.10.1 255.255.255.0
ServerIron10(config)# ip default-gateway 10.10.10.86
ServerIron10(config)# server real R1 10.10.10.2
ServerIron10(config-rs-R1)# port http
ServerIron10(config-rs-R1)# exit
ServerIron10(config)# server real R2 10.10.10.3
ServerIron10(config-rs-R2)# port http
ServerIron10(config-rs-R2)# exit
ServerIron10(config)# server virtual InternalSite10 10.10.10.1

```

```
ServerIron10(config-vs-InternalSite10)# bind http R1 http
ServerIron10(config-vs-InternalSite10)# bind http R2 http
```

### ServerIron 10.10.11.1

```
ServerIron> enable
ServerIron# configure terminal
ServerIron(config)# hostname ServerIron11
ServerIron11(config)# ip address 10.10.11.1 255.255.255.0
ServerIron11(config)# ip default-gateway 10.10.11.86
ServerIron11(config)# server real R3 10.10.11.2
ServerIron11(config-rs-R3)# port http
ServerIron11(config-rs-R3)# exit
ServerIron11(config)# server real R4 10.10.11.3
ServerIron11(config-rs-R4)# port http
ServerIron11(config-rs-R4)# exit
ServerIron11(config)# server virtual InternalSite11 10.10.11.1
ServerIron11(config-vs-InternalSite11)# bind http R3 http
ServerIron11(config-vs-InternalSite11)# bind http R4 http
```

### ServerIron 10.10.12.1

```
ServerIron> enable
ServerIron# configure terminal
ServerIron(config)# hostname ServerIron12
ServerIron12(config)# ip address 10.10.12.1 255.255.255.0
ServerIron12(config)# ip default-gateway 10.10.12.86
ServerIron12(config)# server real R5 10.10.12.2
ServerIron12(config-rs-R5)# port http
ServerIron12(config-rs-R5)# exit
ServerIron12(config)# server real R6 10.10.12.3
ServerIron12(config-rs-R6)# port http
ServerIron12(config-rs-R6)# exit
ServerIron12(config)# server virtual InternalSite12 10.10.12.1
ServerIron12(config-vs-InternalSite12)# bind http R5 http
ServerIron12(config-vs-InternalSite12)# bind http R6 http
```

### ServerIron 192.168.4.1

```
ServerIron> enable
ServerIron# configure terminal
ServerIron(config)# hostname ServerIron69
ServerIron69(config)# ip address 192.168.4.1 255.255.255.0
ServerIron69(config)# ip default-gateway 192.168.4.2
ServerIron69(config)# server real R7 192.168.4.69
ServerIron69(config-rs-R7)# port http
ServerIron69(config-rs-R7)# exit
ServerIron69(config)# server real R8 192.168.4.70
ServerIron69(config-rs-R8)# port http
ServerIron69(config-rs-R8)# exit
ServerIron69(config)# server virtual InternalSite69 192.168.4.69
ServerIron69(config-vs-InternalSite69)# bind http R7 http
ServerIron69(config-vs-InternalSite69)# bind http R8 http
```

## Stateless TCP/UDP Ports

You can configure a TCP application port to be “stateless”. When an application port is stateless, the ServerIron does not create session table entries for the port. Configuring an application port to be stateless provides the following benefits:

- The server responses for the application can use alternate paths back to the client. For example, the

ServerIron and real servers can be connected through a network that provides multiple return paths to the client. Since the port is stateless, the ServerIron does not assume that the application is unhealthy if the server's response does not flow back through the ServerIron.

- The ServerIron has more session resources available for application ports that need them. For example, if your server farm provides non-secure web content in addition to secured transaction processing using SSL, you can use the ServerIron to maintain state information for the SSL connections while allowing the HTTP (web) connections to be stateless. The SSL connections flow back through the ServerIron but the HTTP connections use any available path as determined by a real server's gateway and other routers back to the client.

---

**NOTE:** The SwitchBack feature also allows server responses to take paths that do not pass back through the ServerIron. However, SwitchBack still uses session table resources because the ServerIron creates a session table entry for the connection from the client to the real server.

---

---

**NOTE:** This software release supports stateless TCP/UDP only for stateless application protocols such as HTTP (TCP port 80).

---

## How the ServerIron Selects a Real Server for a Stateless Port

The ServerIron does not use the standard SLB load-balancing methods when selecting a real server for a stateless application port. Instead, the ServerIron uses hash values to select a real server. The ServerIron calculates the hash value for a given client request based on the request's source IP address and source TCP/UDP port.

The ServerIron has 256 hash buckets and divides the 256 buckets evenly among the real servers. When the ServerIron forwards a client's request for a stateless application port to the real server that corresponds to the calculated hash value, the ServerIron does not change the source address of the client's request, but does change the destination address from the requested VIP into the real server's IP address.

For example, when a ServerIron receives a request for TCP port 80 (HTTP) on VIP (192.168.4.69) from client 209.161.1.88, the ServerIron calculates a hash value based on 209.161.1.88 and 80, then forwards the request to the real server that has the calculated hash value. The request packet is in the following format:

- Source IP: client's IP address
- Source application port: port number selected by client's application
- Destination IP: real server's IP
- Destination application port: port number requested by client

If client 209.161.1.88's Web browser sent the request from TCP port 8080, and the ServerIron's hash calculation resulted in selection of real server 10.10.10.2, the packet would have the following address values:

- Source IP: 209.161.1.88
- Source application port: 8080
- Destination IP: real server's IP 10.10.10.2
- Destination application port: 80

Since the client's request contains the client's IP address and application port, the real server can send the packet back to the client by any valid routing path. The request does not need to pass back through the ServerIron that forwarded the request. In fact, the ServerIron that forwards the requests to the transparent VIP does not create session table entries for the requests.

Since the ServerIron does not maintain state information for the requests for the stateless application port, the ServerIron does not care whether the server response for a stateless port passes back through the ServerIron on the way to the client. For a normally configured VIP, the server's response passes back through the ServerIron. For a transparent VIP, the response does not necessarily pass back through the ServerIron.

---

**NOTE:** Since the ServerIron does not create session table entries for requests to the stateless application port, you cannot use ServerIron features that use information in the session table. For example, you cannot use source NAT, port translation, and so on.

---

## Configuring a Stateless Application Port

To configure an application port to be stateless, enable the stateless parameter on the port in the virtual server. Here is an example:

```
ServerIron(config)# server real R1 10.10.10.1
ServerIron(config-rs-R1)# port http
ServerIron(config-rs-R1)# exit
ServerIron(config)# server real R2 10.10.11.1
ServerIron(config-rs-R2)# port http
ServerIron(config-rs-R2)# exit
ServerIron(config)# server virtual StatelessHTTP 192.168.4.69
ServerIron(config-vs-StatelessHTTP)# port http stateless
ServerIron(config-vs-StatelessHTTP)# bind http R1 http
ServerIron(config-vs-StatelessHTTP)# bind http R2 http
```

**Syntax:** [no] port <tcp/udp-portnum> stateless

The <tcp/udp-portnum> parameter specifies the application port you want to make stateless.

---

**NOTE:** This software release supports stateless SLB only for TCP port 80 (HTTP).

---

## Disabling the Stateless SLB Hashing Algorithm for UDP Ports

By default, stateless SLB uses a hashing algorithm to select a real server. The ServerIron calculates a hash value for a given client request based on the request's source IP address and source TCP/UDP port. The request is sent to a real server corresponding to this hash value.

For UDP connections consisting of one client packet and one server response packet, you can disable the stateless SLB hashing algorithm. When the stateless SLB hashing algorithm is disabled for UDP ports, the ServerIron uses the round-robin load balancing method to select a real server for the request. In this case, the ServerIron load balances UDP packets destined for the VIP without creating a session and without calculating hash values based on UDP port number and source IP address.

DNS is an example of a UDP port where this feature can be used. The advantage of disabling the stateless SLB hashing algorithm is that a new real server can be selected immediately after it is brought up.

For example, to disable the stateless SLB hashing algorithm for the DNS port (UDP port 53):

```
ServerIron(config)# server virtual Stateless 192.168.4.69
ServerIron(config-vs-Stateless)# port dns stateless no-hash
```

**Syntax:** [no] port <udp-portnum> stateless no-hash

## Stateless Health Checking

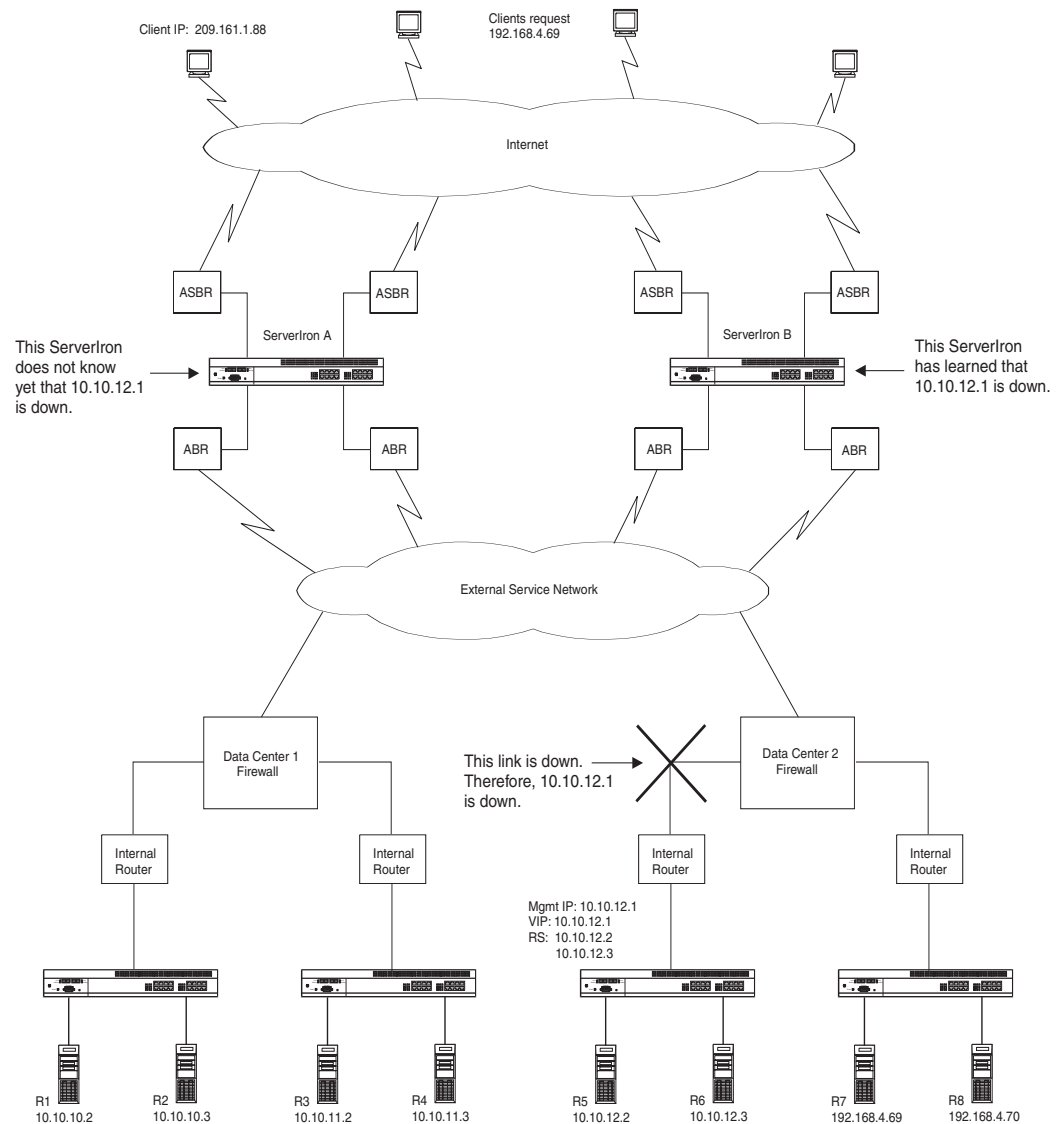
You can configure multiple ServerIrons to coordinate health information for sites that are configured on all the ServerIrons. For example, if you configure a transparent VIP on multiple ServerIrons that have access to the same server farms, stateless health checking ensures that all the ServerIrons share a consistent view of the health of the servers.

Without stateless health checking, it is possible for the ServerIrons to have conflicting health information for a server. For example, if a server goes down, some ServerIrons might know about the down state before others. This can occur due to network congestion or latency, and so on. Since the ServerIrons in a transparent VIP configuration often are on different networks, the ServerIrons that are close to the down server are likely to learn about the server's health change before ServerIrons that are farther away from the server.



Figure 8.3 shows an example of how Serverlrons can have conflicting health check information in a transparent VIP application.

**Figure 8.3 Stateless Health Checking example**



In this example, a link failure has caused 10.10.12.1 to be unavailable. Since transparent VIP uses hashing to select a server, Serverlrons A and B might continue to send requests to 10.10.12.1 until they learn that the site is unavailable. Due to network conditions, Serverlron B learns about the site going down before Serverlron A. As a result, Serverlron A continues to send requests to the down site whereas Serverlron B is already sending the requests to other sites.

Stateless health checking prevents Serverlrons in this type of configuration from having conflicting server health information. To implement stateless health checking, configure a group that contains the management IP addresses of all the Serverlrons configured for transparent VIP. Then assign each of the Serverlrons in the group a stateless health checking priority. The Serverlron with the highest priority becomes the master for server health information. If the master becomes unavailable, the available Serverlron with the highest priority becomes the new master.

## Configuring Stateless Health Checks

To configure stateless health checks:

- Configure the ServerIron group. The group consists of a group ID and a list of the management IP addresses of all the ServerIrons in the group. Configure the same group information on each of the ServerIrons in the group.
- Configure the ServerIron stateless health check priority for the group. The priority determines the master ServerIron for the stateless health check group. In cases where ServerIrons have conflicting information about a real server's state, all the ServerIrons in the group use the state information from the ServerIron with the highest priority.

The following sections describe how to perform these tasks.

---

**NOTE:** In the current release, you can configure only one ServerIron group.

---

### Configuring the ServerIron Group

To configure a stateless health check group, enter a command such as the following on each ServerIron in the group.

```
ServerIronA(config)# server peer-group 1 192.168.3.9 192.168.4.9
```

This command configures group 1 to contain two ServerIrons.

**Syntax:** [no] server peer-group <num> <ip-addr>...

The <num> parameter specifies the stateless health check group ID. You can specify a number from 1 – 16. There is no default.

The <ip-addr>... parameter specifies a list of ServerIron management IP addresses. You can specify up to four addresses with the command. Separate each address with a space. You can configure up to 16 ServerIron management IP addresses. To do so, enter the command four times and specify different addresses each time.

---

**NOTE:** Make sure you add the management IP address for each of the other ServerIrons in the group. Do not include the ServerIron's own management address in the list.

---

### Setting a ServerIron's Stateless Health Check Priority

To configure a ServerIron's stateless health check priority, enter a command such as the following on each ServerIron in the stateless health check group.

---

**NOTE:** If you do not set the stateless health check priority on a ServerIron, that ServerIron does not participate in stateless health checking. If you set the same priority on all the ServerIrons, their priorities are based on their management IP addresses instead. In this case, a higher management IP address has more priority than a lower management IP address.

---

```
ServerIronA(config)# server peer-group 1 self-priority 16
```

This command sets the stateless health check priority on ServerIron A to 16, the highest priority.

**Syntax:** [no] server peer-group <num> <priority>

The <priority> parameter specifies the ServerIron's priority for stateless health checks. You can specify a number from 1 (lowest) – 16 (highest). The ServerIron with the highest stateless health check priority in the group becomes the master for stateless health checks.

To set the priority on ServerIron B, enter a command such as the following:

```
ServerIronB(config)# server peer-group 1 self-priority 1
```

This command sets the stateless health check priority on ServerIron B to 1, the lowest priority.

---

## Chapter 9

# Configuring Global Server Load Balancing

Global Server Load Balancing (GSLB) enables a ServerIron to add intelligence to authoritative Domain Name System (DNS) servers by serving as a proxy to the servers. As a DNS proxy, the GSLB ServerIron evaluates the server IP addresses in the DNS replies from the DNS for which the ServerIron is a proxy. Based on the results of the evaluation, the GSLB ServerIron can change the order of the addresses in the reply so that the “best” host address for the client is on top.

GSLB provides the following advantages:

- No connection delay
- Client geographic awareness based on DNS request origination
- Distributed site performance awareness
- Fair site selection
- Statistical site performance measurements that minimize impact of traffic spikes
- Best performing sites get fair proportion of traffic but are not overwhelmed
- Protection against “best” site failure (HTTP Redirect or IP Proxy as last resort)
- Straight-forward configuration
- All IP protocols are supported

In standard DNS, when a client wants to connect to a host and has the host name but not the IP address, the client can send a lookup request to its local DNS server. The DNS server checks its local database and, if the database contains an Address record for the requested host name, the DNS server sends the IP address for the host name back to the client. The client can then access the host.

If the local DNS server does not have an Address record for the requested server, the local DNS server makes a recursive query. When a request reaches an authoritative DNS server, that DNS server sends a reply to the DNS query. The client’s local DNS server then sends the reply to the client. The client now can access the requested host.

With the introduction of redundant servers, a host name can reside at multiple sites, with different IP addresses. When this is the case, the authoritative DNS server for the host sends multiple IP addresses in its replies to DNS queries. To provide rudimentary load sharing for the addresses, many DNS servers use a simple round-robin algorithm to rotate the list of addresses for each query. Thus, the address that was first in the list in the last reply sent by the DNS server is the last in the list in the next reply sent by the DNS server.

This mechanism can help ensure that a single site for the host does not receive all the requests for the host. However, this mechanism does not provide the host address that is “best” for the client. The best address for the client is the one that has the highest proximity to the client, in terms of being the closest topologically, or

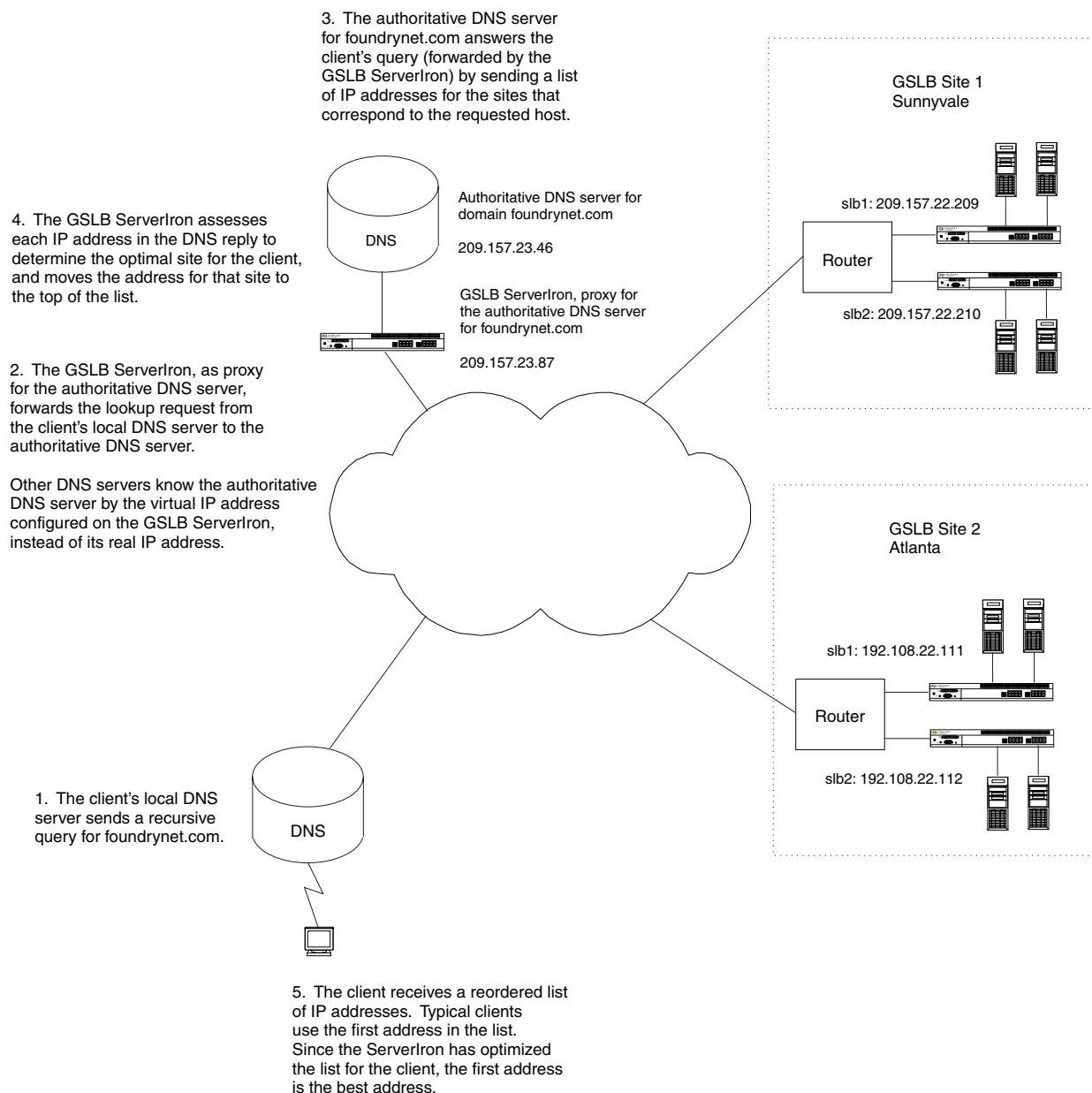
responding the most quickly, and so on. Moreover, if a site is down, the simple round-robin mechanism used by the DNS server cannot tell that the site is down and still sends that site's host address on the top of the list. Thus, the client receives an address for a site that is not available and cannot access the requested host.

The ServerIron GSLB feature solves this problem by intelligently using health checks and other methods to assess the availability and responsiveness of the host sites in the DNS reply, and if necessary exchanging the address at the top of the list with another address selected from the list. Thus, the GSLB feature ensures that a client always receives a DNS reply for a host site that is available and is the best choice among the available hosts.

The rest of this chapter provides an overview of GSLB, configuration procedures, and descriptions of the GSLB information you can display.

## GSLB Overview

Figure 9.1 shows an example of a GSLB configuration. In this example, the GSLB ServerIron (a ServerIron configured for global SLB) is connected to the authoritative DNS server for a specific domain. (You can configure the ServerIron for more than one domain; this example uses only one for simplicity.) The authoritative DNS server for foundrynet.com is known to other devices as 209.157.23.87. This is a VIP configured on the GSLB ServerIron for the DNS server.

**Figure 9.1 Global Server Load Balancing configuration**

This example shows a ServerIron configured as a DNS proxy. The ServerIron is configured as a DNS proxy for the DNS server that is authoritative for the domain foundrynet.com. To configure the ServerIron as a DNS proxy, you identify the DNS name and configure a virtual IP address (VIP) for the DNS. Requests from clients or other DNS servers go to the VIP on the ServerIron, not directly to the DNS server. The ServerIron then sends the requests to the DNS server, transparently to the clients or other DNS servers.

**NOTE:** As an alternative to configuring the GSLB ServerIron as a proxy, you can configure it to intercept and either redirect or directly respond to DNS queries. See "Configuring DNS Cache Proxy" on page 9-47 and "Configuring Transparent DNS Query Intercept" on page 9-50.

The client's local DNS server might cache DNS replies from the authoritative server. Normally, these cached responses would prevent the global SLB from taking place, since the local DNS server would respond directly to the client without sending a recursive query to the authoritative DNS server. However, the GSLB ServerIron, as a

proxy for the authoritative DNS server, automatically resets the Time-to-Live (TTL) parameter in each DNS record from the authoritative server. By default, the GSLB ServerIron sets the TTL to 10 seconds. As a result, other DNS servers that receive the records retain them in their databases for only 10 seconds. After the ten seconds expire, subsequent requests from the client initiate another query to the authoritative DNS server. As a result, the client always receives fresh information and the address of the site that is truly the best site for the client.

---

**NOTE:** You also can change the TTL if needed. However, Foundry Networks recommends that you do not change the TTL to 0, because this can be interpreted as an error by some older DNS servers.

---

You identify each ServerIron by its management IP address, not by any VIPs configured on the ServerIron. Optionally, you also can specify a name for each ServerIron at the site.

If a remote site is managed by one or more ServerIrons, the GSLB ServerIron gathers additional information from the site ServerIrons using Foundry's GSLB protocol with the remote ServerIrons. The protocol uses TCP port 182. To initiate the GSLB protocol between the GSLB ServerIron and the ServerIrons at the remote sites, you must first enable the GSLB protocol on those remote ServerIrons, then identify the sites and the ServerIrons. In this example, the GSLB ServerIron is configured with site information for Site 1 in Sunnyvale and Site 2 in Atlanta. Each site has servers containing the content for domain names within the domain foundrynet.com. The servers are load balanced by the ServerIrons.

The GSLB protocol is disabled by default. You must enable the GSLB protocol on each site ServerIron. After you enable the GSLB protocol, the GSLB ServerIron finds the site ServerIrons using their IP management addresses, which you specify when you configure the remote site information.

The GSLB ServerIron uses the GSLB protocol to learn the following information from the site ServerIrons:

- The VIPs configured on the site ServerIrons – The site ServerIrons report VIP additions and deletions asynchronously. Each time a VIP is added to a site ServerIron, the ServerIron sends a message to the GSLB ServerIron to inform the GSLB ServerIron of the change.
- Session table statistics and CPU load information – The site ServerIrons report this information to the GSLB ServerIron at regular intervals. By default, each remote ServerIron sends the status information to the GSLB ServerIron every 30 seconds. You can change the update period for all the remote ServerIrons by specifying a new period on the GSLB ServerIron if needed.
- Round-trip time (RTT) – The Round-trip time (RTT) is the amount of time that passes between when the remote site receives a TCP connection (TCP SYN) from the client and when the remote site receives the client's acknowledgment of the connection request (TCP ACK). The GSLB ServerIron learns the RTT information from the site ServerIrons through the Foundry GSLB protocol and uses the information as a metric when comparing site IP addresses.
- (Optional) Connection load – A GSLB site's connection load is the average number of new connections per second on the site, over a given number of intervals. When you enable this GSLB metric, all potential candidates are compared against a predefined load limit. All sites that have fewer average connections than the threshold are selected and passed to the next comparison metric. The connection load metric is disabled by default but is enabled (added to the GSLB policy) when you configure the metric.

---

**NOTE:** All the ServerIrons in the GSLB configuration (the GSLB ServerIron and the remote site ServerIron) must be running the same software release.

---

The GSLB ServerIron uses the information supplied by the GSLB protocol when comparing the sites and may re-order the IP addresses in the authoritative DNS server's reply based on the results of the comparison. If you have enabled the GSLB protocol on the site ServerIrons, the GSLB ServerIron begins communicating with the site ServerIrons using the GSLB protocol as soon as you add the site definitions to the GSLB ServerIron.

When you configure the GSLB ServerIron, you also specify the zones for which you want the ServerIron to provide global SLB. These are the zones for which the DNS server (the one the ServerIron is a proxy for) is the authority. In this example, the DNS server is an authority for foundrynet.com. Only the zones and host names you specify receive global SLB. The DNS server can contain other host names that are not globally load balanced or otherwise managed by the GSLB ServerIron.

You also must specify the host names and applications that you want to provide global SLB for. For example, assume that foundrynet.com contains the following host names and applications:

Host Name	Application
-----------	-------------

www.foundrynet.com	HTTP
--------------------	------

ftp.foundrynet.com	FTP
--------------------	-----

The application specifies the type of health check the GSLB ServerIron applies to IP addresses for the host. A host name can be associated with more than one application. In this case, the GSLB ServerIron considers a host name's IP address to be healthy only if the address passes all the health checks. The ServerIron has Layer 7 health checks for the following applications.

- FTP – the well-known name for port 21. (Ports 20 and 21 both are FTP ports but on the ServerIron, the name corresponds to port 21.)
- TFTP – the well-known name for port 69
- HTTP – the well-known name for port 80
- IMAP4 – the well-known name for port 143
- LDAP – the well-known name for port 389
- NNTP – the well-known name for port 119
- POP3 – the well-known name for port 110
- SMTP – the well-known name for port 25
- TELNET – the well-known name for port 23

---

**NOTE:** To display the list when configuring zone information, enter the **host-info** <host-name> ? command, where <host-name> is a string specifying a host name.

---

For other applications (applications not listed above), the ServerIron does not perform a Layer 7 health check but still performs a Layer 4 TCP or UDP health check.

You can customize the HTTP health check on an individual host basis by changing the URL string the ServerIron requests in the health check and the list of HTTP status codes the ServerIron accepts as valid responses to the health check.

The Layer 4 and Layer 7 health checks are described in detail in “Configuring Port and Health Check Parameters” on page 12-1.

## The GSLB Policy

The ServerIron can use the following metrics to evaluate the server IP addresses in a DNS reply:

- The server's health
- The site ServerIron's session capacity threshold
- The round-trip time between the remote ServerIron and the DNS client's sub-net
- The geographic location of the server
- The connection load
- The site ServerIron's available session capacity
- The site ServerIron's FlashBack speed (how quickly the GSLB receives the health check results)
- The site ServerIron's administrative preference (a numeric preference value you assign to influence the GSLB policy if other policy metrics are equal)
- The Least Response selection (the site ServerIron that has been selected less often than others)
- Round robin selection (an alternative to the Least Response metric)

---

**NOTE:** The default order for the metrics is the order shown above.

---

The GSLB ServerIron evaluates each IP address in the DNS reply based on these metrics. Based on the results, the GSLB ServerIron can reorder the list to place the IP address for the “best” site on the top of the list.

If the GSLB policy rejects all of the sites, the GSLB ServerIron sends the DNS reply unchanged to the client.

All of these metrics have default values but you can change the values if needed. In addition, you can disable individual metrics or reorder them. See “Modifying the GSLB Policy Parameters” on page 9-33.

You also can configure the GSLB ServerIron to directly respond to DNS queries instead of forwarding the queries to the authoritative DNS server and modifying the replies. See “Configuring DNS Cache Proxy” on page 9-47.

The following sections describe each of these metrics in detail.

### Server Health

The GSLB ServerIron sends a Layer 4 TCP or UDP health check and Layer 7 application health check to the server to determine the health of the server and the host application on the server. If the server fails either health check, the GSLB ServerIron immediately disqualifies the server’s IP address from being the “best” site.

The GSLB ServerIron determines which health checks to use based on the host applications you specify. For example, if a host name is associated with both HTTP and FTP applications, the ServerIron sends the site Layer 4 TCP health checks (one for HTTP and one for FTP) and also sends a separate Layer 7 HTTP health check and a separate Layer 7 FTP health check. The site must pass all the health checks or it is disqualified from being the best site.

If a host application uses a port number that is not known to the ServerIron and supported by GSLB, the ServerIron cannot perform a Layer 7 health check on the application but still performs a Layer 4 TCP or UDP health check on the port. Health check parameters such as retry interval, number of retries, and so on are global parameters.

The Layer 4 and Layer 7 health checks and procedures for configuring them are described in detail in “Configuring Port and Health Check Parameters” on page 12-1.

---

**NOTE:** You can change the order in which the GSLB policy applies the metrics. However, Foundry Networks recommends that you always use the health check as the first metric. Otherwise, it is possible that the GSLB policy will not select a “best” choice, and thus send the DNS reply unchanged. For example, if the first metric is geographic location, and the DNS reply contains two sites, one in North America and the other in South America, the GSLB policy favors the South American site after the first comparison. However, if that site is down, the GSLB policy will find that none of the sites in the reply is the “best” one, and thus send the reply unchanged.

---

---

**NOTE:** If all the sites fail their health checks, resulting in all the sites being rejected by the GSLB ServerIron, the ServerIron sends the DNS reply unchanged to the client.

---

### Site ServerIron’s Session Capacity Threshold

The GSLB protocol supplies statistics for the session tables on each site ServerIron. The session table contains an entry for each open TCP or UDP session on the site ServerIron. Each ServerIron has a maximum number of sessions that it can hold in its session table. Through the Foundry GSLB protocol, the GSLB ServerIron learns from each remote ServerIron the maximum number of sessions and the number of available sessions on that ServerIron.

The capacity threshold specifies how close to the maximum session capacity the site ServerIron (remote ServerIron) can be and still be eligible as the best site for the client. This mechanism provides a way to shift load away from a site before the site becomes congested.

The default value for the threshold is 90%. Thus a site ServerIron is eligible to be the best site only if its session utilization is below 90%. See “Displaying GSLB Information” on page 9-55 for commands to display a site’s utilization and the capacity threshold.



## Round-Trip Time Between the Remote ServerIron and the Client

The Round-trip time (RTT) is the amount of time that passes between when the remote site receives a TCP connection (TCP SYN) from the client and when the remote site receives the client's acknowledgment of the connection request (TCP ACK). The GSLB ServerIron learns the RTT information from the site ServerIrons through the Foundry GSLB protocol and uses the information as a metric when comparing site IP addresses.

The GSLB ServerIron maintains a database of cache entries, which contains the information about past DNS queries. The information is aggregated on a network-address prefix basis. When the GSLB ServerIron receives a DNS query, it creates or updates a cache entry. Round-trip Time (RTT) measurements reported by remote ServerIrons are then sorted into the cache. The GSLB ServerIron uses this information for decisions on subsequent DNS queries. If a cache entry is not refreshed for a while (there are no subsequent queries from the same address prefix), the ServerIron clears the entry from the RTT database.

When the GSLB ServerIron compares two site IP addresses based on RTT, the GSLB ServerIron favors one site over the other only if the difference between the RTT values is greater than the specified percentage. This percentage is the RTT tolerance. You can set the RTT tolerance to a value from 0 – 100. The default is 10%.

Site ServerIrons send RTT information only for the sessions that clients open with them. To prevent the GSLB ServerIron from biasing its selection toward the first site ServerIron that sent RTT information, the GSLB ServerIron intentionally ignores the RTT metric for a specified percentage of the requests from a given client network. You can specify an RTT explore percentage from 0 – 100. The default is 5. By default, the GSLB ServerIron ignores the RTT for 5% of the client requests from a given network.

To configure RTT parameters, see “Modifying Round-Trip Time Values” on page 9-41.

## Geographic Location of the Server

For each client query, the GSLB ServerIron can determine the geographic location from which the client query came based on its IP address. The GSLB can determine whether the query came from North America, Asia, Europe, or South America. If multiple sites compare equally based on the metrics above, the GSLB ServerIron prefers sites within the same geographic region as the client query.

---

**NOTE:** The GSLB ServerIron deduces the geographic region of the client's local DNS server from the destination IP address in the DNS reply, which is the address of the client's local DNS server.

---

The GSLB ServerIron determines the geographic region of a server IP address in its DNS database in the following ways:

- For real IP addresses (as opposed to VIPs, which are logical IP addresses configured on the site ServerIrons), the geographic region is based on the IP address itself.
- For VIPs, the geographic region is based on the management IP address of the site ServerIron on which the VIP is configured.
- You can explicitly specify the region if the management IP address of the remote ServerIron is not indicative of the geographic location. For example, if the management IP address is in a private sub-net, the address does not indicate the ServerIron's geographic location. If you specify the region, the ServerIron uses the region you specify instead of the region of the ServerIron's management IP address.

## Site ServerIron's Connection Load

A GSLB site's connection load is the average number of new connections per second on the site, over a given number of intervals. When you enable this GSLB metric, all potential candidates are compared against a predefined load limit. All sites that have fewer average connections than the threshold are selected and passed to the next comparison metric. The connection limit metric is disabled by default but is enabled (added to the GSLB policy) when you configure the metric.

---

**NOTE:** This metric is supported in software release 07.2.25 or later and 07.3.04 or later.

---

## Site ServerIron's Available Session Capacity Tolerance

If multiple sites are equal after the above comparisons, the GSLB ServerIron prefers the site ServerIron (remote ServerIron) whose session table has the most unused entries.

When comparing sites based on the session table utilization, the GSLB ServerIron considers the sites to be equal if the difference in session table utilization does not exceed the tolerance percentage. The tolerance percentage ensures that minor differences in utilization do not cause frequent, and unnecessary, changes in site preference.

For example, suppose one ServerIron has 1 million sessions available, and another has 800,000 sessions available. Also assume that the tolerance is 10% (the default). In this case the first ServerIron (with 1 million sessions available) is preferred over the second ServerIron because the difference (200,000) is greater than 10% of 1 million. If a third ServerIron has 950,000 sessions available, that ServerIron is equally preferable with the first ServerIron (with 1 million sessions available), because the difference in percentage between the available sessions on the two ServerIrons is only 5%, which is less than the tolerance threshold.

### Site ServerIron's FlashBack Speed

If multiple sites compare equally based on all the metrics above, the ServerIron chooses a site as the best one based on how quickly the GSLB ServerIron received responses to health checks to the site ServerIron.

The GSLB ServerIron uses a tolerance value when comparing the FlashBack speeds of different sites. The tolerance value specifies the percentage by which the FlashBack speeds of the two sites must differ in order for the ServerIron to choose one over the other. The default FlashBack tolerance is 10%. Thus, if the FlashBack speeds of two sites are within 10% of one another, the ServerIron considers the sites to be equal. However, if the speeds differ by more than 10%, the ServerIron prefers the site with the lower FlashBack speed.

FlashBack speeds are measured at Layer 4 for all TCP/UDP ports. For the application ports known to the ServerIron, the FlashBack speed of the application is also measured.

When the ServerIron compares the FlashBack speeds, it compares the Layer 7 (application-level) FlashBack speeds first, if applicable. If the application has a Layer 7 health check and if the FlashBack speeds are not equal, the ServerIron is through comparing the FlashBack speeds. If a host is associated with multiple applications, the GSLB ServerIron uses the slowest response time among the applications for the comparison. However, if only the Layer 4 health check applies to the application, or if further tie-breaking is needed, the ServerIron then compares the Layer 4 FlashBack speeds.

### Site ServerIron's Administrative Preference

The administrative preference is an optional metric. This metric is a numeric preference value from 0 – 255 that you assign to each site ServerIron, to select that ServerIron if the previous metrics do not result in selection of a best site. The GSLB policy prefers the site ServerIron with the highest administrative preference.

The administrative preference allows you to do the following:

- You can temporarily change the preference of a site to accommodate changing network conditions. For example, if sites are offering proxy content service, the link between a site proxy server farm and the content origin may be highly congested, making that site less desirable. This factor is not visible to the ServerIrons and thus cannot be reflected in the other GSLB metrics.
- You can temporarily disqualify a site ServerIron from being selected, without otherwise changing the site's configuration or the GSLB ServerIron's configuration. For example, you can perform maintenance on the site ServerIron without making network changes. In this case, set the administrative preference to 0.
- You can bias a GSLB ServerIron that is also configured as a site ServerIron (for locally configured VIPs) to always favor itself as the best site. In this case, assign an administrative preference of 255 to the site for the GSLB ServerIron itself, and assign a lower administrative distance to the other site ServerIrons, or use the default (128) for those sites.

The administrative preference is disabled by default, which means it is not included as one of the GSLB metrics. When you enable this metric, the default administrative preference for sites is 128. You can change the preference on an individual site basis. To change a site's preference, see "Specifying the GSLB Sites and the Site ServerIrons" on page 9-19.

### The Least Response Selection

If multiple sites still compare equally based on all the metrics above, the GSLB ServerIron selects the site that it has selected least often before. For example, if the GSLB ServerIron has selected Site 1 and placed its IP address on top in 40% of the DNS replies, but has selected Site 2 60% of the time, then in this instance the GSLB

ServerIron selects Site 1. To display the response selection percentages for the sites you have configured, use the **show gslb dns zone** command. See “Displaying Zone and Host Name Information” on page 9-62.

This metric is a tie-breaker in case multiple addresses pass through all the above comparisons without one address emerging as the best choice. If this occurs, the address of the site that has been selected least often in previous DNS responses is selected.

Least Response Selection is enabled by default. You can disable the metric only by enabling the Round Robin Selection metric to act as the tie breaker instead. See the following section.

### Round Robin Selection

The Round Robin Selection metric is an alternative to the Least Response Selection metric as the final tie breaker. When you enable Round Robin Selection, the GSLB ServerIron automatically disables the Least Response Selection metric, and instead uses the round-robin algorithm to select a site. Round Robin Selection chooses the first IP address in the DNS response for the first client request, then selects the next address for the next client request, and so on.

Use the Round Robin Selection metric instead of the Least Response Selection metric when you want to prevent the GSLB ServerIron from favoring new or recently recovered sites over previously configured active sites. The Least Response metric can cause the GSLB ServerIron to select a new site or a previously unavailable site that has come up again instead of previously configured sites for a given VIP. This occurs because the GSLB ServerIron has selected the new site fewer times than previously configured sites for the VIP.

In some cases, the Least Response Selection metric can cause the GSLB ServerIron to send client requests to a new or recovered site faster than the site can handle while it is coming up. To avoid this situation, you can configure the GSLB ServerIron to use the Round Robin Selection metric instead of the Least Response Selection metric as the final tie breaker.

The Round Robin Selection metric is disabled by default.

## Configuring GSLB

To configure GSLB, perform the following tasks. The examples in the procedures in this section are based on the configuration shown in Figure 9.1 on page 9-3.

**Table 9.1: Configuration tasks – Global SLB**

Feature	See page...
<b>Configure DNS Proxy Parameters</b>	
Configure a source IP address. The source IP address is required so that the GSLB ServerIron can perform the health checks on remote devices.	9-11
Add a real-server definition for the DNS server.	
Add a VIP for the DNS server and bind the real server and virtual server.	
<b>Configure Site Parameters</b>	
Enable the GSLB protocol on each remote ServerIron.	9-18
Specify the sites and the ServerIrons within the sites.	9-19
<b>Configure Zone Parameters</b>	
Specify the zones and the host names within the zones.	9-21
<b>Modify GSLB Parameters (optional)</b>	
Change the GSLB protocol port number (optional).	9-27

**Table 9.1: Configuration tasks – Global SLB (Continued)**

Feature	See page...
Change the GSLB protocol update period (optional).	9-28
Remove IP addresses for sites that fail a Layer 4 or application health check.	9-29
Remove all IP addresses except the “best” one from the DNS reply.	9-30
Change the refresh interval for zone and host information.	9-30
Change the TTL value the ServerIron sets for the DNS records.	9-31
Disable TTL modification	9-31
Re-order the GSLB policy metrics	9-35
Disable health checks	9-36
Change the session table capacity and threshold tolerance	9-39
Change the FlashBack tolerance	9-40
Change Round-trip time (RTT) parameters	9-41
<b>Configure Affinity (optional)</b>	
Configure the ServerIron to always favor a specific site based on client IP address	9-44
<b>Configure DNS Cache Proxy (optional)</b>	
Configure the ServerIron to directly respond to DNS queries	9-47
<b>Configure Transparent DNS Query Intercept (optional)</b>	
Configure the ServerIron to intercept and redirect DNS queries	9-50
<b>Disable or Re-enable GSLB Traps (optional)</b>	
Disable or re-enable GSLB SNMP traps and Syslog messages	9-77

You can configure the GSLB ServerIron to be a proxy for more than one DNS server.

As shown in the example in Figure 9.1 on page 9-3, you implement GSLB by connecting a ServerIron to an authoritative DNS server. To configure the ServerIron for GSLB, perform the following steps:

- Add the proxy information for the DNS server. To configure the GSLB ServerIron as a proxy for the DNS server, add real server definition for the DNS server, then add a virtual server (VIP) for the DNS server and bind the real and virtual servers.

**NOTE:** The virtual server IP address (VIP) will be the Authoritative DNS server for the GSLB Domain.

- If a site contains ServerIrons, identify the server sites. A server site is a data center or server farm connected to the Internet by a router. This example shows two GSLB sites. Each of the sites is connected to the Internet by a router.
- If a site contains ServerIrons, identify the ServerIrons within the server sites. This initiates the Foundry GSLB protocol between the ServerIron acting as a DNS proxy and the remote ServerIrons in the GSLB sites. The DNS proxy uses information supplied by the remote ServerIrons to assess the preferability of IP addresses in the DNS replies.

**NOTE:** You can use the GSLB ServerIron for standard SLB. In this case, identify the local site and the GSLB ServerIron in the same way as you identify the other sites and ServerIrons. The configuration steps are the same.

---

- Identify the DNS zones and the hosts within those zones for which you want the GSLB ServerIron to perform GSLB. You must specify the zones and hosts. There are no defaults.
- Identify the host applications with each host. The GSLB ServerIron performs GSLB for the applications you specify. You can specify applications known to the ServerIron as well as the TCP or UDP port numbers of applications that are not known to the ServerIron. The ServerIron performs Layer 7 and Layer 4 health checks for the applications known to the ServerIron, but performs only Layer 4 health checks for applications that are not known to it. See “Server Health” on page 9-6.

## Configuring the Proxy

To configure the GSLB ServerIron as a proxy for a DNS server:

- If the GSLB ServerIron and site ServerIrons are in different sub-nets, add a source IP address. In this case, the source IP address is required so that the GSLB ServerIron perform the health checks on the IP addresses the GSLB ServerIron learns from the DNS server for which it is the proxy. The source IP address must be in the same sub-net as the GSLB ServerIron’s management IP address.

**NOTE:** You can specify as many DNS servers as the GSLB ServerIron’s system memory allows. However, the ServerIron sends periodic DNS queries to only the first four DNS servers you configure with the DNS proxy.

If you configure the ServerIron as a proxy for multiple DNS servers, make sure they have identical content for the zones that you configure the GSLB ServerIron to provide GSLB services for.

---

- Add a real server for the DNS server.
- Add a virtual server for the DNS server and bind the real DNS server and virtual server together.

## Adding a Source IP Address

To enable the GSLB ServerIron to perform health checks on remote sites that are in a sub-net other than the GSLB ServerIron’s sub-net, you must add a source IP address to the GSLB ServerIron. The source IP address must be in the same sub-net as the GSLB ServerIron’s management IP address.

**NOTE:** If the DNS server for which the GSLB ServerIron is a proxy is in a different sub-net than the GSLB ServerIron’s management IP address, you can use the same source IP address that you add for the site ServerIrons. However, you also need to enable the Source NAT feature for the DNS real server.

---

The source IP address and source NAT feature allow the ServerIron to send a Layer 4 or Layer 7 health check to the remote site and receive the response. Notice that the source IP address added to the ServerIron is not in the sub-net of the remote ServerIron. Instead, the source IP address is in the sub-net that connects the ServerIron’s local router to the Internet. The purpose of the source IP address in this configuration is to ensure that the responses from remote sites come back to the ServerIron. The health check packets use the address you configure as their source IP address. Without the source IP address in the ServerIron’s sub-net and the source feature, the responses to the health checks sent to remote sites in different sub-nets cannot reach the ServerIron.

For example, the GSLB ServerIron shown in Figure 9.1 on page 9-3 needs a source IP address in the sub-net 209.157.23.x. Without this source IP address, Layer 4 and Layer 7 health checks to the ServerIrons at the Sunnyvale site (209.157.22.x) and the Atlanta site (192.108.22.x) cannot reach the GSLB ServerIron.

To add a source IP address, use either of the following methods.

### USING THE CLI

To add a source IP address, enter a command such as the following:

```
ServerIron(config)# server source-ip 209.157.23.225 255.255.255.0 0.0.0.0
```

**Syntax:** [no] server source-ip <ip-addr> <ip-mask> <default-gateway>

The <ip-addr> parameter specifies the IP address. Specify an address that is in the same sub-net as the GSLB ServerIron's management IP address. Do not specify an address that is already in use.

The <ip-mask> parameter specifies the network mask.

The <default-gateway> parameter specifies the default gateway. This parameter is required, but if you do not want to specify a gateway, enter "0.0.0.0".

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the Source IP link from the menu. The Source IP panel is displayed, as shown in the following example.

**Source IP**

<b>IP Address:</b>	209.157.23.225
<b>Subnet Mask:</b>	255.255.255.0
<b>Default Gateway:</b>	0.0.0.0

[\[Show\]](#)

[\[Home\]](#)
[\[Site Map\]](#)
[\[Logout\]](#)
[\[Save\]](#)
[\[Frame Enable\]](#)
[\[Disable\]](#)
[\[TELNET\]](#)

---

**NOTE:** If you have already configured source IP address information, a table listing the source IP addresses is displayed. Click the Add Source IP link to add a new one or select the Modify button next to the one you want to modify.

---

5. Enter the IP address in the IP Address field.
6. Enter the network mask in the Subnet Mask field.
7. Optionally, enter a default gateway in the Default Gateway field, or leave "0.0.0.0" in the field.
8. Click Add (if you are adding a new one) or Modify (if you are modifying one that was already configured) to implement the change.
9. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

#### Configuring a Real Server and Virtual Server for the DNS Server

To add a real server and virtual server for the DNS server and bind them together, use either of the following methods.

---

**NOTE:** The virtual server IP address (VIP) will be the Authoritative DNS server for the GSLB Domain.

---

#### USING THE CLI

To configure a real server and virtual server and bind them together for a proxy DNS server, enter commands such as the following:

```
ServerIron(config)# server real-name dns_ns 209.157.23.46
ServerIron(config-rs-dns_ns)# port dns proxy
```

```

ServerIron(config-rs-dns_ns)# exit
ServerIron(config)# server virtual-name dns-proxy 209.157.23.87
ServerIron(config-vs-dns-proxy)# port dns
ServerIron(config-vs-dns-proxy)# bind dns dns_ns dns

```

The commands in this example add a real server called “dns\_ns”. The DNS server has IP address 209.157.23.46. When you add the real server, the CLI changes to the Real Server configuration level. At this level, you can add TCP or UDP ports and, optionally, modify health check parameters. In this example, the DNS port is added. Notice that the **proxy** option is specified following the **dns** option. The **proxy** option is required to indicate that this real server is part of a proxy DNS server configuration.

If the DNS server is in a different sub-net than the GSLB ServerIron, you must configure a source IP address on the ServerIron for use by the health checks. If the GSLB ServerIron is in a one-armed configuration or the DNS server is at least one hop away, you must configure a source IP address and also enable source NAT. (You do not need to add another source IP address if you have already added one for the remote sites. The GSLB ServerIron can use the same source IP address for reaching the remote sites and for reaching the DNS server.)

```

ServerIron(config)# server real-name dns_ns 209.157.23.46
ServerIron(config-rs-dns_ns)# port dns proxy
ServerIron(config-rs-dns_ns)# exit

```

The **server virtual-name** command adds a virtual server called “dns-proxy”. This command changes the CLI to the Virtual Server configuration level. At this level, the **port dns** command adds the DNS port to the virtual server. The **bind** command binds the DNS port on the real server to the DNS port on the virtual server.

**Syntax:** [no] server real-name <text> <ip-addr>

**Syntax:** [no] port dns proxy

**Syntax:** [no] port <port> [disable | enable]

**Syntax:** [no] port <port> [keepalive]

**Syntax:** [no] server virtual-name <text> [<ip-addr>]

**Syntax:** [no] bind <port> <real-server-name> <port>

For information about the health check parameters, see “Configuring Port and Health Check Parameters” on page 12-1.

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the Real Server link from the menu to display the Real Server panel, as shown in Figure 9.2.

**Figure 9.2 Real Server panel**

**Real Server**

Server Name:	<input type="text" value="dns_ns"/>
Server IP:	<input type="text" value="209.157.23.46"/>
Maximum Connections:	<input type="text" value="1000000"/>
Weight:	<input type="text" value="1"/>
Host Range:	<input type="text" value="1"/>
Remote:	<input type="checkbox"/>
Source NAT:	<input type="checkbox"/>

[\[Show\]](#)

[\[Virtual Server\]](#)
[\[Virtual Server Port\]](#)
[\[Real Server Port\]](#)
[\[Bind\]](#)

[\[Home\]](#)
[\[Site Map\]](#)
[\[Logout\]](#)
[\[Save\]](#)
[\[Frame Enable\]](#)
[\[Disable\]](#)
[\[TELNET\]](#)

---

**NOTE:** If you have already configured real server information, a table listing the real servers appears. Select the [Add Real Server](#) link under the table to add a new real server or select the Modify button next to the row of information about the real server you want to modify.

---

5. Enter a name for the server in the Server Name field. this name is used for ServerIron configuration and does not need to match any specific information on the DNS server itself.
6. Enter the DNS server IP address in the Server IP field.
7. If the DNS server is in a different sub-net than the ServerIron's management IP address and you added a source IP address, select the checkbox next to Source NAT to enable this feature.

---

**NOTE:** For information about the Maximum Connections, Weight, and Host Range fields, see "Configuring Server Load Balancing" on page 6-1. You can configure GSLB without changing the values in these fields.

---

8. Click Add (if you are adding a new server) or Modify (if you are modifying a server that was already configured) to implement the change.
9. Repeat steps 5 – 8 for up to three additional DNS servers.
10. Select the [Real Server Port](#) link to display the Real Server Port panel, as shown in Figure 9.3.



Figure 9.3 Real Server Port panel

**Real Server Port**

Server Name:	dns_ns	
TCP/UDP Port:	DNS	User Define
Status:	<input type="radio"/> Disable <input checked="" type="radio"/> Enable	
Keep Alive:	<input checked="" type="checkbox"/>	
<b>DNS Parameters</b>		
+DNS Zone:		
+Addr Query:		
+Proxy:	<input type="checkbox"/>	
<b>HTTP Parameters</b>		
*Method:	HEAD	
*URL:		
*Status Code:		
<b>Group Id Range</b>		
From	To	
<input type="text"/>	<input type="text"/>	
<input type="text"/>	<input type="text"/>	
<input type="text"/>	<input type="text"/>	
<input type="text"/>	<input type="text"/>	

**NOTE:** If you have already configured real server port information, a table listing the real server port configurations appears. Select the [Add Real Server Port](#) link under the table to add a new real server port or select the Modify button next to the row of information about the real server port you want to modify.

11. Select the name of the real server you added above from the Server Name field's pulldown menu.
12. Select DNS from the TCP/UDP Port field's pulldown menu.
13. Select the Keep Alive checkbox to enable keepalive messages for the DNS port.
14. Select the Proxy checkbox in the DNS Parameters section. You must select this option to indicate that the port is for a DNS server proxy.

**NOTE:** For information about the other fields in the DNS Parameters section, see "Configuring the DNS Health Check Method and Values" on page 12-38. These fields let you customize the Layer 7 DNS health check for this DNS server.

15. Click Add (if you are adding a new port) or Modify (if you are modifying a server port that was already configured) to implement the change.
16. Select the [Virtual Server](#) link from the bottom of the Real Server Port panel to display the Virtual Server panel, as shown in Figure 9.4.

**Figure 9.4 Virtual Server panel**

**Virtual Server**

Server Name:	<input type="text" value="dns-proxy"/>
Server IP:	<input type="text" value="209.157.23.87"/>
Host Range:	<input type="text" value="1"/>
Symmetric Priority:	<input type="text" value="0"/>
HTTP Redirect:	<input type="checkbox"/>
Load Balancing Metric:	<input checked="" type="radio"/> Default <input type="radio"/> Least Connection <input type="radio"/> Round Robin <input type="radio"/> Weighted

[\[Show Virtual Server\]](#)

[\[Track\]](#)
[\[Virtual Server Port\]](#)
[\[Real Server\]](#)
[\[Real Server Port\]](#)
[\[Bind\]](#)

[\[Home\]](#)
[\[Site Map\]](#)
[\[Logout\]](#)
[\[Save\]](#)
[\[Frame Enable\]](#)
[\[Disable\]](#)
[\[TELNET\]](#)

---

**NOTE:** If you have already configured virtual server information, a table listing the virtual servers appears. Select the [Add Virtual Server](#) link under the table to add a new virtual server or select the Modify button next to the row of information about the virtual server you want to modify.

---

17. Enter the virtual server name in the Server Name field.
18. Enter the virtual IP address (VIP) for the DNS server. This is the proxy address, the address by which other DNS servers will know this DNS server.

---

**NOTE:** For information about the Host Range, Symmetric Priority, HTTP Redirect, and Load Balancing Metric fields, see “Configuring Server Load Balancing” on page 6-1. You can configure GSLB without changing the values in these fields.

---

19. Click Add (if you are adding a new virtual server) or Modify (if you are modifying a virtual server that was already configured) to implement the change.
20. Select the [Virtual Server Port](#) link to display the Virtual Server Port panel, as shown in Figure 9.5.

Figure 9.5 Virtual Server Port panel

**Virtual Server Port**

Server Name:	dns-proxy
TCP/UDP Port:	DNS <input type="button" value="User Define"/>
Status:	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Sticky:	<input type="checkbox"/>
Concurrent:	<input type="checkbox"/>
Translate:	<input checked="" type="checkbox"/>
DSR:	<input type="checkbox"/>
<b>HTTP Parameters</b>	
URL Switching:	<input type="checkbox"/>
URL Map:	None
Cookie Switching:	<input type="checkbox"/>
Cookie Name:	
<b>SSL Parameter</b>	
Session-id Switching:	<input type="checkbox"/>

[Show]

[Host Id][URL Map][Track][Virtual Server][Real Server][Real Server Port][Bind]

[Home][Site Map][Logout][Save][Frame Enable][Disable][TELNET]

**NOTE:** If you have already configured virtual server port information, a table listing the virtual server port configurations appears. Select the [Add Virtual Server Port](#) link under the table to add a new virtual server port or select the Modify button next to the row of information about the virtual server port you want to modify.

21. Select the name of the virtual server you added above from the Server Name field's pulldown menu.
22. Select DNS from the TCP/UDP Port field's pulldown menu.

**NOTE:** For information about the other fields on this panel, see "Configuring Global Server Load Balancing" on page 9-1. These fields are not required for configuring GSLB.

23. Click Add (if you are adding a new port) or Modify (if you are modifying a server port that was already configured) to implement the change.
24. Select the [Bind](#) link from the bottom of the Virtual Server Port panel to display the Bind panel, as shown in Figure 9.6.

**Figure 9.6 Bind panel**

**Bind**

Virtual Server Name:	dns-proxy
Virtual TCP/UDP Port:	DNS
Real Server Name	dns_ns
Real TCP/UDP Port:	DNS

[Show]

[\[Virtual Server\]](#)
[\[Virtual Server Port\]](#)
[\[Real Server\]](#)
[\[Real Server Port\]](#)

[\[Home\]](#)
[\[Site Map\]](#)
[\[Logout\]](#)
[\[Save\]](#)
[\[Frame Enable\]](#)
[\[Disable\]](#)
[\[TELNET\]](#)

---

**NOTE:** If you have already configured port bindings, a table listing the bindings appears. Select the [Add Bind](#) link under the table to add a new binding.

---

25. Select the virtual server from the Virtual Server Name field's pulldown menu.
26. Select DNS from the Virtual TCP/UDP Port field's pulldown menu.
27. Select the real server from the Real Server Name field's pulldown menu.
28. Select DNS from the Real TCP/UDP Port field's pulldown menu.
29. Select the Bind button to add the port binding.
30. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Enabling the GSLB Protocol On the Site ServerIrons

For security, remote ServerIrons do not listen to TCP port 182 (the GSLB protocol port) by default. This means the GSLB protocol is disabled on remote site ServerIrons by default. For a remote ServerIron to use the protocol, you must enable the protocol on the remote ServerIron.

---

**NOTE:** Enter this command on the site ServerIron, not on the GSLB ServerIron.

---



---

**NOTE:** The ServerIron uses TCP port 182 for the GSLB protocol by default. You can change the port number if needed. See "Changing the GSLB Protocol Port Number" on page 9-27.

---



---

**NOTE:** You also can secure access to a ServerIron by configuring Access Control Lists (ACLs). For example, you can configure ACLs to control access to the device on TCP port 182. See the "Using Access Control Lists (ACLs)" chapter in the *Foundry Enterprise Configuration and Management Guide*.

---

To enable a remote ServerIron to use the GSLB protocol (TCP port 182), use the following CLI method on the remote ServerIron.

### USING THE CLI

To enable a remote ServerIron to use the GSLB protocol, enter the following command:

```
ServerIron(config)# gslb protocol
```

**Syntax:** [no] gslb protocol

## USING THE WEB MANAGEMENT INTERFACE

You cannot use the Web management interface to enable the GSLB protocol.

## Specifying the GSLB Sites and the Site ServerIrons

To identify the SLB sites, you create a site definition and identify the ServerIrons in the site.

---

**NOTE:** If the GSLB ServerIron itself is also a site for a host, the configuration steps are the same as for remote site ServerIrons. Add a site definition and then specify the GSLB ServerIron as the ServerIron at the site. Specify the management IP address as the ServerIron IP address.

---



---

**NOTE:** If traffic between the GSLB ServerIron and a remote site ServerIron must pass through a firewall, the firewall must be configured to allow traffic to and from the GSLB ServerIron.

---

To identify a server site and the ServerIron(s) within it, use either of the following methods.

### USING THE CLI

To identify the server sites shown in Figure 9.1 on page 9-3, enter the following commands:

```
ServerIron(config)# gslb site sunnyvale
ServerIron(config-gslb-site-sunnyvale)# si-name slb-1 209.157.22.209
ServerIron(config-gslb-site-sunnyvale)# si-name slb-2 209.157.22.210
ServerIron(config)# gslb site atlanta
ServerIron(config-gslb-site-atlanta)# si-name slb-1 192.108.22.111
ServerIron(config-gslb-site-atlanta)# si-name slb-2 192.108.22.112
```

These commands configure two GSLB sites. One of the sites is in Sunnyvale and the other is in Atlanta. Each site contains two ServerIrons that load balance traffic across server farms. The GSLB ServerIron you are configuring will use information provided by the other ServerIrons when it evaluates the servers listed in DNS replies.

**Syntax:** [no] gslb site <name>

The <name> parameter is a text string that uniquely identifies the site on the GSLB ServerIron. You can enter a string up to 16 characters long. The string can contain blanks. To use blanks, enclose the string in quotation marks.

---

**NOTE:** If you delete a GSLB site (by entering the **no gslb site <name>** command), the site and all the ServerIrons you associated with the site are deleted.

---

**Syntax:** [no] si-name [<name>] <ip-addr> [<preference>]

The <name> parameter specifies a unique name for the ServerIron at the site. You can enter a string up to 16 characters long. The string can contain blanks. To use blanks, enclose the string in quotation marks. You can enter up to four pairs of ServerIron names and IP addresses on the same command line. The name is optional.

---

**NOTE:** Enter the ServerIron's management IP address, not a virtual IP address (VIP) configured on the ServerIron or a source IP address added for source NAT.

---

The <preference> parameter sets the administrative preference for the site. When you enable the administrative preference as a GSLB metric, the administrative preference can be used by the GSLB policy when comparing this site with other sites. You can specify a preference from 0 – 255. The default preference is 128. The GSLB policy prefers high preference values over low preference values. If you specify 0, the site is administratively removed from selection by the GSLB policy but remains connected to the network. See “Site ServerIron's Administrative Preference” on page 9-8 for information about uses for this parameter.

For example, to set the administrative preference for a site ServerIron to 255, enter a command such as the following:

```
ServerIron(config-gslb-site-sunnyvale)# si-name slb-1 209.157.22.20 255
```

To change the preference for a site ServerIron you have already configured, use the same command syntax. You do not need to reconfigure other site parameters when you change the preference. For example, to change the preference for a site ServerIron from the default (128) to 200, enter a command such as the following:

```
ServerIron(config-gslb-site-sunnyvale)# si-name slb-2 209.157.22.210 200
```

**NOTE:** The administrative preference metric is disabled by default, which means it is not used by the GSLB policy. The GSLB policy uses the preference values only if you enable this metric. See “Disabling or Enabling GSLB Policy Metrics” on page 9-36.

By default, the GSLB ServerIron uses a site’s IP address to determine its geographic location. Alternatively, you can explicitly identify the location. To do so, use the following command.

**Syntax:** [no] geo-location asia | europe | n-america | s-america

For example, to explicitly identify Sunnyvale’s geographic location as North America, enter the following commands:

```
ServerIron(config)# gslb site sunnyvale
ServerIron(config-gslb-site-sunnyvale)# geo-location n-america
```

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the [Site](#) link to display the GSLB Site panel, as shown in Figure 9.7.

**Figure 9.7** GSLB Site panel

**GSLB Site**

<b>Name:</b>	<input type="text" value="Sunnyvale"/>		
<b>Geographic Location:</b>	<input type="text" value="North America"/>		
<b>ServerIron</b>			
<b>Name</b>	<b>IP</b>	<b>Administrative Preference</b>	
<input type="text" value="slb-1"/>	<input type="text" value="209.157.22.209"/>	<input type="text" value="128"/>	
<input type="text" value="slb-2"/>	<input type="text" value="209.157.22.210"/>	<input type="text" value="128"/>	
<input type="text" value=""/>	<input type="text" value="0.0.0.0"/>	<input type="text" value="128"/>	
<input type="text" value=""/>	<input type="text" value="0.0.0.0"/>	<input type="text" value="128"/>	

[\[Show\]](#)[\[DNS\]](#)[\[Policy\]](#)

[\[Home\]](#)[\[Site Map\]](#)[\[Logout\]](#)[\[Save\]](#)[\[Frame Enable\]](#)[\[Disable\]](#)[\[TELNET\]](#)

**NOTE:** If you have already configured remote site information, a table listing the sites appears. Select the [Add](#) link under the table to add a new site or select the [Modify](#) button next to the row of information about a site you want to change.

5. Enter the site name in the Name field. You can enter a string up to 16 characters long. The string can contain blanks. To use blanks, enclose the string in quotation marks.
6. If the site is not in North America, select a geographic location for the site from the Geographic Location field’s pulldown menu. The GSLB ServerIron uses the location information when evaluating the addresses in DNS replies. See “Geographic Location of the Server” on page 9-7.

7. Enter the name of a ServerIron at the site in the Name column under ServerIron.
8. Enter the site ServerIron's management IP address in the IP column.
9. Optionally, edit the ServerIron's administrative preference.
10. Repeat steps 7 – 8 for up to four ServerIrons at this site.
11. Click Add (if you are adding a new site) or Modify (if you are modifying a site that was already configured) to implement the change.
12. Repeat steps 5 – 11 for each site.
13. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Specifying the DNS Zones and the Host Applications

You must specify the DNS zones and the host applications within each zone for which you want the GSLB ServerIron to provide global SLB. There are no defaults for these parameters.

As soon as you specify the hosts and applications, the GSLB ServerIron queries the DNS server (the one for which the GSLB ServerIron is a proxy) for the IP addresses associated with the hosts and begins sending health checks to the hosts.

To specify the zones and host applications, use either of the following methods.

### USING THE CLI

To specify the foundrynet.com zone and two host names, each of which is associated with an application, enter the following commands:

```
ServerIron(config)# gslb dns zone-name foundrynet.com
ServerIron(config-gslb-dns-foundrynet.com)# host-info www http
ServerIron(config-gslb-dns-foundrynet.com)# host-info ftp ftp
```

The commands in this example add the zone foundrynet.com and add two hosts within that zone: www and ftp. The GSLB ServerIron will provide global SLB for these two hosts within the zone.

**Syntax:** [no] gslb dns zone-name <name>

The <name> parameter specifies the DNS zone name.

---

**NOTE:** If you delete a DNS zone (by entering the **no gslb dns zone-name** <name> command), the zone and all the host names you associated with the zone are deleted.

---

**Syntax:** [no] host-info <host-name> <host-application> | <TCP/UDP-port-num>

The <host-name> parameter specifies the host name. You do not need to enter the entire (fully-qualified) host name. Enter only the host portion of the name. For example, if the fully qualified host name is www.foundrynet.com, do not enter the entire name. Enter only "www". The rest of the name is already specified by the **gslb dns zone-name** command. You can enter a name up to 32 characters long.

The <host-application> specifies the host application for which you want the GSLB ServerIron to provide global SLB. You can specify one of the following:

- FTP – the well-known name for port 21. (Ports 20 and 21 both are FTP ports but on the ServerIron, the name "FTP" corresponds to port 21.)
- TFTP – the well-known name for port 69
- HTTP – the well-known name for port 80
- IMAP4 – the well-known name for port 143
- LDAP – the well-known name for port 389
- NNTP – the well-known name for port 119

- POP3 – the well-known name for port 110
- SMTP – the well-known name for port 25
- TELNET – the well-known name for port 23

The <TCP/UDP-port-num> parameter specifies a TCP/UDP port number instead of a well-known port. If the application is not one of those listed above, you still can configure the GSLB ServerIron to perform the Layer 4 health check on the specified port.

---

**NOTE:** If the application number does not correspond to one of the well-known ports recognized by the ServerIron, the GSLB ServerIron performs Layer 4 TCP or UDP health checks for the ports but does not perform application-specific health checks.

---

### USING THE WEB MANAGEMENT INTERFACE

---

**NOTE:** The HTTP Parameters fields are supported only in ServerIron 400 and ServerIron 800 software release 07.1.x and higher. The fields are not supported in the 07.1.x releases of other ServerIron products.

---

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the DNS link to display the GSLB DNS panel, as shown in Figure 9.8.

**Figure 9.8** GSLB DNS panel

**GSLB DNS**

Zone Name:	foundrynet.com	
Host Name:	www	
TCP/UDP Port:	HTTP	User Define
HTTP Parameters		
*Method:	HEAD	
*URL:		
*Status Code:		

[\[Show\]](#)
[\[IP List\]](#)
[\[Policy\]](#)
[\[Site\]](#)
[\[Real Server\]](#)
[\[Real Server Port\]](#)

[\[Home\]](#)
[\[Site Map\]](#)
[\[Logout\]](#)
[\[Save\]](#)
[\[Frame Enable\]](#)
[\[Disable\]](#)
[\[TELNET\]](#)

---

**NOTE:** If you have already configured zone and host information, a table listing the information appears. Select the Add link under the table to add a new zone.

---

5. Enter the zone name in the Zone Name field.
6. Enter the host name in the Host Name field.
7. Select or enter the TCP or UDP application port for an application on the host you specified above.
  - To select a TCP or UDP application port known to the ServerIron, select the port from the field's pulldown menu.
  - To enter a port number that is not known to the ServerIron, click the User Define button to change the pulldown menu into an entry field, then enter the TCP or UDP port number.



8. Optionally, edit the health check parameters for HTTP. For information about HTTP health check parameters, see “Modifying HTTP Health Check Parameters” on page 9-23.
  - Select GET or HEAD from the Method pulldown menu.
  - Specify a URL that the health check will request.
  - Specify the status codes the ServerIron will accept as valid responses to the health check.
9. Click Add to implement the change.
10. Repeat steps 7 – 9 for each application on this host.
11. Repeat steps 6 – 9 for each host.
12. Repeat steps 5 – 9 for each zone.
13. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device’s flash memory.

### Support for Canonical Name (CNAME) DNS Records

GSLB supports CNAME records. If you configure domain names that map to other domain names, the GSLB ServerIron still will perform GSLB for the domain. A CNAME record is a type of DNS record that allows network administrators to create aliases for domain names. For example, an administrator can create the following DNS records for the Foundry Networks domain:

- Address record: www.foundrynet.com, IP address 209.157.22.241
- CNAME record: www.foundrynetworks.com, alias for www.foundrynet.com

A CNAME record refers to another domain name instead of an IP address. A client can enter either domain name to get to the site. Each domain name goes to site 209.157.22.241. However, in software release 06.0.03, the ServerIron did not support CNAME records in DNS responses. As a result, if you configured the GSLB ServerIron to provide GSLB for a domain name that mapped to another domain name instead of to one or more IP addresses, the GSLB ServerIron did not learn the addresses properly.

### Modifying HTTP Health Check Parameters

For HTTP hosts, you also can customize the health check by changing the URL method and the string requested by the ServerIron, as well as the HTTP status codes the ServerIron accepts as valid responses. By default, the ServerIron performs the HTTP health check as follows:

- The ServerIron sends a HEAD request for the default URL string, “HEAD /”.
- If the server responds with the status code 401 or a code in the range 200 – 299, the server passes the health check.

You can change the request method from HEAD to GET. In addition, you can change the URL string the ServerIron requests from the server and the status codes that the ServerIron accepts as valid responses for passing the health check.

The commands in the following example change the method from HEAD to GET and to add 404 as a valid status code response to the health check.

```
ServerIron(config)# gslb dns zone foundrynet.com
ServerIron(config-gslb-dns-foundrynet.com)# host-info www http url "GET /index.htm"
ServerIron(config-gslb-dns-foundrynet.com)# host-info www http status-code 200 299
401 401 404 404
```

**Syntax:** host-info <host-name> http | <TCP-portnum> url “[GET | HEAD] [/]<URL-page-name>”

**GET** or **HEAD** is an optional parameter that specifies the request type. By default, HTTP keepalive uses HEAD to retrieve the URL page. You can override the default and configure the ServerIron to use GET to retrieve the URL page.

The slash ( / ) is an optional parameter. If you do not set the GET or HEAD parameter, and the slash is not in the configured URL page, then ServerIron automatically inserts a slash before retrieving the URL page.

**Syntax:** host-info <host-name> http | <TCP-portnum> status-code <range> [<range> [<range> [<range>]]]

You can specify up to four ranges (total of eight values). To specify a single message code for a range, enter the code twice. For example to specify 200 only, enter the following command: **port http status-code 200 200.**

---

**NOTE:** When you change the status code ranges, the defaults are removed. As a result, you must specify all the valid ranges, even if a range also is within the default ranges. For example, if you still want codes 200 – 299 to be valid, you must specify them.

---



---

**NOTE:** When a URL string is associated with a TCP port number rather than the well-known HTTP port, the ServerIron performs both a TCP and an HTTP health check. In this case, you must specify the method and URL before specifying the status code ranges. The software displays an error message if you accidentally try to change the status codes before specifying the method and URL.

---

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the DNS link to display a table listing the configured GSLB DNS zone and host information.
5. Select Modify next to the row for the host and zone you want to modify.
6. Edit the health check parameters for HTTP. For information about HTTP health check parameters, see “Modifying HTTP Health Check Parameters” on page 9-23.
  - Select GET or HEAD from the Method pulldown menu.
  - Specify a URL that the health check will request.
  - Specify the status codes the ServerIron will accept as valid responses to the health check.
7. Click Modify to implement the change.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device’s flash memory.

### Configuring DNS Domain Name Aliases

You can configure an alias for a domain name and application configured on the GSLB ServerIron. This feature is useful together with the DNS cache proxy feature when you want the GSLB ServerIron to learn a set of proxy server IP addresses for a domain, then respond to client requests with the best proxy server address.

Typically, you use this set of features when the DNS server contains a single server address for the actual domain name, but a separate set of proxy server addresses for an alias for that domain name. When you enable DNS cache proxy and configure the alias for the domain on the GSLB ServerIron, the GSLB ServerIron:

- Learns the proxy server addresses under the alias on the DNS server instead of the address for the domain’s actual site. This requires configuration of the alias on the GSLB ServerIron.
- Responds to client queries for the actual domain name with the best site address from among the proxy server addresses learned from the DNS server under the alias. This requires that enable the DNS cache proxy feature.

---

**NOTE:** Use this feature only in conjunction with the DNS cache proxy feature. Otherwise, it is possible for the IP address(es) the ServerIron learns under the real domain name and the addresses it learns under the alias to be different. When this is the case, the ServerIron does not make any alterations to the DNS response but instead sends the response back to the client unaltered. As a result, the ServerIron sends the client the DNS reply with the real domain name’s server address, instead of one of the proxy addresses configured on the DNS server under the domain’s alias.

---

To configure an alias for a domain name, use the following CLI method.

#### *USING THE CLI*

To configure an alias for a domain name, enter the alias after entering the zone name and host application names, as shown in the following example.

```
ServerIron(config)# gslb dns zone foundrynet.com
ServerIron(config-gslb-dns-foundrynet.com)# host www http
ServerIron(config-gslb-dns-foundrynet.com)# host www alias www.gslb.foundrynet.com
```

**Syntax:** host-info <host-name> alias <alias-name>

The commands in this example configure a zone called “foundrynet.com”, associate an HTTP host named “www” with the zone, then associate the alias “www.gslb.foundrynet.com” with the host and zone.

---

**NOTE:** Make sure you configure the alias only after configuring the zone and the host application the alias is for, as shown in the example above. In addition, make sure you specify the fully-qualified name for the alias (for example, “www.gslb.foundrynet.com” instead of “www.gslb”).

---

#### *USING THE WEB MANAGEMENT INTERFACE*

You cannot configure a DNS alias using the Web management interface.

### **Support for Null Host Names**

When you configure a zone name in GSLB, you enter the zone name, then associate host applications with the zone name. For example, you might configure the following for the “foundrynet.com” zone:

- www.foundrynet.com (HTTP application)
- ftp.foundrynet.com (FTP application)

Some e-commerce sites also accept just a zone name as an alias for a specific application within that zone. For example, a site might accept both “www.foundrynet.com” and “foundrynet.com” as valid names for the HTTP application on the web host. In this case, the second name has a null host name. No application is explicitly associated with the “foundrynet.com” zone, but the DNS server is configured to associate “foundrynet.com” with the same IP address(es) and application as “www.foundrynet.com”, for example using address records or alias records.

---

**NOTE:** The real Authoritative DNS server must be configured to support Null Host.

---

To configure a null host name on the GSLB ServerIron, use the following CLI method.

#### *USING THE CLI*

To configure a null host name, enter commands such as the following:

```
ServerIron(config)# gslb dns zone-name foundrynet.com
ServerIron(config-gslb-dns-foundrynet.com)# host-info www http
ServerIron(config-gslb-dns-foundrynet.com)# host-info null-host http
```

The last command in the example above configures a null host for the foundrynet.com zone and associates the null host with HTTP.

**Syntax:** [no] host-info <host-name> | null-host <host-application> | <TCP/UDP-port-num>

You can configure one null host for each application and zone name.

#### *USING THE WEB MANAGEMENT INTERFACE*

You cannot configure a null host name using the Web management interface.

### **Configuration Example**

Here is a complete proxy server configuration example for a GSLB ServerIron. In addition to the commands shown above, this example contains the commands for configuring the ServerIron as a DNS server proxy.

```
ServerIron(config)# server real-name dns_ns 192.10.10.1
ServerIron(config-rs-dns_ns)# port dns proxy
ServerIron(config-rs-dns_ns)# exit
ServerIron(config)# server virtual-name dns-proxy 192.10.10.69
ServerIron(config-vs-dns-proxy)# port dns
ServerIron(config-vs-dns-proxy)# bind dns dns_ns dns
```

The commands above configure the GSLB ServerIron as the proxy for the client's DNS server. The following commands configure the zone and host information for foundrynet.com and specify the IP address of the proxy server. When the ServerIron receives a reply from the client's DNS server for foundrynet.com, the ServerIron replaces the IP address in the reply with 209.157.23.59, the IP address of a proxy server.

```
ServerIron(config)# gslb dns zone foundrynet.com
ServerIron(config-gslb-dns-foundrynet.com)# host www http
ServerIron(config-gslb-dns-foundrynet.com)# host www ip-list 209.157.23.59
```

The following commands enable DNS override on the ServerIron. DNS override allows the ServerIron to replace the IP address in the DNS reply with the IP address you configure for the proxy server.

```
ServerIron(config-vs-dns-proxy)# exit
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# dns override
```

You must enable DNS override for the ServerIron to replace the address. Otherwise, the ServerIron still uses the GSLB policy to select a "best" site but does not replace the IP address with the proxy server's address.

#### **USING THE WEB MANAGEMENT INTERFACE**

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the Policy link to display the GSLB Policy panel, as shown in Figure 9.9 on page 9-27.
5. Click on the checkbox next to Override, in the DNS section of the panel, to place a checkmark in the box.
6. Click Apply to implement the change.
7. Click on the IP List link to display the GSLB DNS IP List panel.
8. Select the DNS host name from the Host field's pulldown list.
9. Enter up to eight IP addresses in the IP address fields, then click Add to add them.
10. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## **Modifying GSLB Protocol Parameters**

This section describes how to modify the following GSLB protocol parameters.

- GSLB protocol port number – changes the port number of the GSLB protocol from TCP port 182 to another port number
- GSLB protocol update period – specifies how often the site ServerIrons report their session table statistics and CPU utilization to the GSLB ServerIron. The default update period is 30 seconds. You can change the period to a value from 1 – 300 seconds.
- DNS response parameters:
  - Effect of health-check results on IP addresses in DNS reply
  - DNS record verification interval
  - TTL value for DNS records
  - DNS override

- GSLB policy parameters:
  - Metric processing order – you can change the order in which the metrics are applied.
  - Metric state – you can disable or re-enable some metrics.
  - Session-table capacity and threshold tolerance – you can modify the values for these metrics.
  - FlashBack tolerance – you can modify the value for this metric.
  - RTT values – you can individually modify the cache interval, cache prefix, tolerance, and explore percentage.
  - Connection load parameters – you can adjust the number of data collection intervals and the relative weights given to the intervals.

CLI and Web management procedures are provided. To modify GSLB parameters using the Web management interface, use the GSLB Policy panel, shown in Figure 9.9.

**Figure 9.9** GSLB Policy panel

**GSLB Policy**

Protocol Status Interval:		30
<b>DNS</b>		
Check Interval:		30
TTL:		10
Active Server Only:		<input type="checkbox"/>
Best Only:		<input type="checkbox"/>
Override:		<input type="checkbox"/>
<b>Metric Order</b>		
Order #	Enable	Type
1	<input checked="" type="checkbox"/>	Health Check
2	<input checked="" type="checkbox"/>	Session Capacity Threshold
		Threshold: 90
3	<input checked="" type="checkbox"/>	Round Trip Time
		Cache Interval: 120
		Explore Percentage: 5
		Prefix Length: 20
		RTT Tolerance: 10
4	<input checked="" type="checkbox"/>	Geographic Location
5	<input checked="" type="checkbox"/>	Available Session Capacity
		Tolerance: 10
6	<input checked="" type="checkbox"/>	Flashback
		Application Tolerance: 10
		TCP Tolerance: 10
7	<input checked="" type="checkbox"/>	Administrative Preference

## Changing the GSLB Protocol Port Number

By default, a GSLB ServerIron uses TCP port 182 to exchange GSLB information with other ServerIrons, including the site ServerIrons. You can change the GSLB protocol port if needed. For example, if other devices in the network also use port 182, but for other applications, you need to change the protocol on those devices or on the ServerIrons.

**NOTE:** If you change the GSLB protocol port number, you must save the change to the startup-config file and reload the software to place the change into effect. Also, you must change the port to the same number on all ServerIrons in the GSLB configuration. If the port number in two GSLB ServerIrons is not the same, those ServerIrons are not able to properly perform GSLB.

---

To change the GSLB protocol port number on a ServerIron, enter commands such as the following:

```
ServerIron(config)# gslb communication 1882
ServerIron(config)# write memory
ServerIron(config)# end
ServerIron# reload
```

The first command changes the TCP protocol port from 182 to the specified port number, in this example 1882. The subsequent commands save the configuration change to the startup-config file and reload the software to place the change into effect.

**Syntax:** [no] gslb communication <tcp-portnum>

The <tcp-portnum> parameter specifies the TCP port number you want the ServerIron to use for exchanging GSLB information with other ServerIrons.

## Changing the GSLB Protocol Update Period

By default, each remote ServerIron uses the GSLB protocol to send status information to the GSLB ServerIron every 30 seconds. The status information consists of session utilization and CPU load information, which you can display using the **show gslb site** command (see “Displaying Site Information” on page 9-56).

You can change the update period if needed. The valid range is 1 – 300 seconds. To change the update period, specify a new period on the GSLB ServerIron. The GSLB ServerIron then informs all the remote ServerIrons of the change.

### USING THE CLI

To change the GSLB protocol update period, enter the following commands on the GSLB ServerIron:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# protocol status-interval 10
```

The command in this example changes the GSLB protocol update period to 10 seconds.

**Syntax:** [no] protocol status-interval <num>

The <num> parameter specifies the number of seconds between status reports and can be from 1 – 300 seconds. The default is 30 seconds.

To display the current update period, enter the **show gslb policy** command. The interval is shown in the Remote SI status update period field. See “Displaying the GSLB Policy” on page 9-67 for more information.

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the Policy link to display the GSLB Policy panel, as shown in Figure 9.9 on page 9-27.
5. Edit the value in the Protocol Status Interval field. You can enter a value from 1 – 300 seconds. The default is 30 seconds.
6. Click Apply to implement the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device’s flash memory.

## Modifying the GSLB Parameters Related to DNS Responses

You can modify the following DNS-related GSLB parameters:

- Treatment of IP address for a site that fails a health check – By default, the ServerIron does not remove an IP address from a DNS reply even if the address fails a health check. You can configure the ServerIron to remove such addresses from the DNS responses.
- Treatment of IP addresses that pass the health checks but still are not selected as the “best” site – By default, the ServerIron leaves all the IP addresses in the DNS reply. You can configure the ServerIron to remove all addresses from the reply except the “best” address.
- Frequency with which the ServerIron verifies its current DNS records with DNS servers – As soon as you add site and host information for GSLB, the ServerIron sends DNS queries to the DNS server (the one for which the ServerIron is the proxy) to get the IP addresses associated with the zones and host names you specified. After this, the ServerIron refreshes this information by sending new DNS queries every 30 seconds. You can change the query interval.
- TTL value the ServerIron sets for the DNS records – By default, the ServerIron sets the TTL to 10 seconds in the DNS records in all the replies from the DNS server for which the ServerIron is performing GSLB. The TTL controls how long other DNS servers, including the client’s DNS server, keep the query results in their databases. You can change this TTL.

If you prefer to manage the TTL values solely using the DNS server, you can disable TTL modification on the ServerIron.

---

**NOTE:** Foundry Networks recommends that you do not change the TTL to 0, because this can be interpreted as an error by some older DNS servers.

---

- DNS override – By default, the GSLB ServerIron selects the best site IP address from among the addresses contained in the DNS reply. You can override the DNS reply for an individual zone and host by specifying a list of IP addresses, then enabling DNS override. Thus configured, the GSLB ServerIron replaces the DNS response using the best address from among a list of addresses you associate with the host. This feature is useful when you want to provide the best address for a web proxy without the need to configure the proxy’s IP address onto the DNS server itself.

To change the DNS parameters, use the following methods.

### Deleting IP Addresses for Sites that Fail a Health Check

You can configure the ServerIron to remove IP addresses from DNS replies when those addresses fail a health check. The ServerIron removes the addresses that fail the check so long as the DNS query still contains at least one address that passes the health check.

A site must pass all applicable health checks (Layer 4 and Layer 7) to avoid being removed.

---

**NOTE:** If all the sites fail their health checks, resulting in all the sites being rejected by the GSLB ServerIron, the ServerIron sends the DNS reply unchanged to the client.

---

#### USING THE CLI

To configure the ServerIron to remove IP addresses from DNS replies when those addresses fail a health check, enter the following commands.

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# dns active-only
```

**Syntax:** [no] dns active-only

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.

3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the Policy link to display the GSLB Policy panel, as shown in Figure 9.9 on page 9-27.
5. Click on the checkbox next to Active Server Only to enable the feature.
6. Click Apply to implement the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Deleting All IP Addresses Except the Best One

By default, the GSLB ServerIron retains the same number of IP addresses in the DNS replies from the DNS server. The GSLB policy swaps the IP address on the top of the list with the “best” address, selected by the GSLB policy. You can configure the ServerIron to remove all addresses except the one the GSLB policy selects as the best address.

---

**NOTE:** If the GSLB policy does not result in the selection of a “best” address, the DNS reply can still contain multiple addresses.

---

To do so, use either of the following methods.

#### USING THE CLI

To configure the GSLB ServerIron to remove all addresses except the best address from the DNS replies, enter the following commands:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# dns best-only
```

**Syntax:** [no] dns best-only

To display the state of this feature, enter the **show gslb policy** command. The DNS best-only field indicates whether the feature is enabled or disabled. See “Displaying the GSLB Policy” on page 9-67.

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the Policy link to display the GSLB Policy panel, as shown in Figure 9.9 on page 9-27.
5. Click on the checkbox next to Best Only, in the DNS section of the panel, to place a checkmark in the box.
6. Click Apply to implement the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Changing the Refresh Interval for Zone and Host Information

The GSLB ServerIron periodically sends DNS queries to verify the zone and host information. The GSLB ServerIron sends the queries to the DNS server for which it is configured to be a proxy. The default interval is 30 seconds. You can change the interval to a value from 0 – 1000000000 seconds.

#### USING THE CLI

To change the refresh interval, enter commands such as the following:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# dns check-interval 50
```

**Syntax:** [no] dns check-interval <num>

The <num> parameter specifies the interval and can be from 1 – 1000000000 seconds. The default is 30 seconds.



### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the Policy link to display the GSLB Policy panel, as shown in Figure 9.9 on page 9-27.
5. Edit the value in the Check Interval field, in the DNS section of the panel, to a value from 1 – 1000000000 seconds. The default is 30 seconds.
6. Click Apply to implement the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Changing the TTL For Records in the DNS Reply

The GSLB ServerIron changes the TTL of each DNS record contained in the DNS replies from the DNS server for which the ServerIron is a proxy. By default, the GSLB ServerIron changes the TTL to 10. You can modify this value to from 0 – 1000000000 seconds.

### USING THE CLI

To change the TTL, enter commands such as the following:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# dns ttl 45
```

**Syntax:** [no] dns ttl <num>

The <num> parameter specifies the TTL and can be from 0 – 1000000000 seconds. The default is 10 seconds.

For all GSLB features except DNS cache proxy, the command **no dns ttl** configures the ServerIron to use the TTL from the DNS server. If you are using DNS cache proxy, this command resets the TTL to 10.

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the Policy link to display the GSLB Policy panel, as shown in Figure 9.9 on page 9-27.
5. Edit the value in the TTL field to a value from 0 – 1000000000 seconds. The default is 10 seconds.
6. Click Apply to implement the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Disabling TTL Modification

If you prefer to manage the TTL values solely on the DNS server and do not want the ServerIron to modify the TTL, you can disable TTL modification. To do so, enter the following command:

```
ServerIron(config-gslb-policy)# no dns ttl
```

**Syntax:** [no] dns ttl

### USING THE WEB MANAGEMENT INTERFACE

You cannot disable TTL modification using the Web management interface.

### Enabling DNS Override

By default, the GSLB ServerIron selects the best site IP address from among the addresses contained in the DNS reply. You can override the DNS reply for an individual domain (zone plus a host) by specifying a list of IP

addresses, then enabling DNS override. Thus configured, the GSLB ServerIron replaces the DNS response using the best address from among the list of IP addresses you associate with the host.

This feature is useful when you want to provide the best address for a web proxy without the need to configure the proxy's IP address onto the DNS server itself.

DNS override is a global parameter. You configure redirection on an individual host basis, then globally enable the GSLB ServerIron to replace the IP addresses in the DNS reply with the proxy server addresses you configure. Once you configure DNS override, for each domain name (zone and host) configured on the GSLB ServerIron, there must be a set of IP addresses configured for the domain. The GSLB ServerIron replaces the IP addresses in a DNS response with the best choice (only the best choice) from the set of configured IP addresses. If a domain name does not have a configured address, the ServerIron sends the DNS reply unaltered to the client.

---

**NOTE:** The host and its associated health check (if applicable) must be configured before you configure the IP address list.

---

You can specify as many proxy server IP addresses as you need for a given domain. When you specify multiple proxy server addresses, the ServerIron uses the applicable GSLB policy metrics to select the best address from the list of addresses you configure and places that address in the DNS reply.

#### *USING THE CLI*

To configure the proxy server information on the GSLB ServerIron, enter commands such as the following:

```
ServerIron(config)# gslb dns zone foundrynet.com
ServerIron(config-gslb-dns-foundrynet.com)# host www http
ServerIron(config-gslb-dns-foundrynet.com)# host www ip-list 209.157.23.59
```

**Syntax:** host <host-name> ip-list <ip-addr...>

The <host-name> parameter specifies the host name.

The **ip-list** <ip-addr...> specifies the proxy IP address(es). You can specify as many proxy IP addresses as you need. If you specify multiple addresses, separate each address with a space. Here is an example:

```
host www ip-list 209.157.23.59 209.157.23.60 209.157.23.61 207.142.33.6
```

For information about the other syntax for the **host** command, see “Specifying the DNS Zones and the Host Applications” on page 9-21.

To enable DNS override, enter the following command. You must enable DNS override to allow the ServerIron to insert the proxy IP address in the DNS reply.

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# dns override
```

**Syntax:** [no] dns override

When you enable DNS override, the GSLB ServerIron replaces the IP addresses in the DNS reply with the “best” of the proxy server addresses you specify. The GSLB ServerIron determines which proxy server IP address is the best using the GSLB policy metrics. For information about the metrics, see “The GSLB Policy” on page 9-5.

---

**NOTE:** DNS override is a global parameter but a list of proxy IP addresses are associated with a specific host in a specific domain. If there are no proxy addresses for a given host, the GSLB ServerIron sends the DNS reply unaltered. An exception is if you have enabled the active only feature, in which case the reply contains only the active addresses. See “Deleting IP Addresses for Sites that Fail a Health Check” on page 9-29.

---

To display the DNS override state, enter the **show gslb policy** command. The state is shown in the DNS override field. See “Displaying the GSLB Policy” on page 9-67 for more information.

To display information about the IP addresses selected for a specific domain and host, enter the **show gslb dns zone** command. See “Displaying Zone and Host Name Information” on page 9-62.

## Modifying the GSLB Policy Parameters

“The GSLB Policy” on page 9-5 describes the default policy the GSLB ServerIron uses to evaluate the IP addresses in the DNS replies from the DNS server for which the ServerIron is configured as a proxy. You can change the policy by changing or deleting individual metrics. Table 9.2 lists the GSLB policy metrics. The metrics are listed in their default order. The metric described in the first row is the first metric the GSLB ServerIron uses by default, and so on.

**NOTE:** If the GSLB policy rejects all of the sites, the GSLB ServerIron sends the DNS reply unchanged to the client.

**Table 9.2: GSLB Policy Metrics**

metric	Default	Configuration Options
Server (host) health	<p>Enabled.</p> <p>The GSLB ServerIron performs Layer 4 health checks on the TCP or UDP port and Layer 7 health checks on the application, if the application is known to the ServerIron.</p> <p><b>Note:</b> If all the sites fail their health checks, resulting in all the sites being rejected by the GSLB ServerIron, the ServerIron sends the DNS reply unchanged to the client.</p>	<p>You can disable this metric.</p> <p><b>Note:</b> When both the health check metric and the Flashback metric are disabled, the ServerIron does not perform any Layer 4 or Layer 7 health checks.</p>
Session capacity threshold	<p>Enabled.</p> <p>The default value for the threshold is 90%. Thus a site ServerIron is eligible to be the best site only if its session utilization is below 90%.</p>	<p>You can change the threshold to a value from 0 – 100%.</p> <p>You can disable this metric.</p>
Round-trip time (RTT) from remote ServerIrons to the DNS clients.	<p>Enabled.</p> <p>The default RTT cache interval is 120 seconds.</p> <p>The default cache prefix length is 20 bits.</p> <p>The default tolerance (used when comparing otherwise equal sites) is 10%.</p> <p>The default explore percentage is 5%.</p>	<p>You can change the RTT cache interval, cache prefix length, tolerance, and explore percentage individually.</p> <p>You can disable this metric. If you disable RTT, the GSLB ServerIron instructs the site ServerIrons to stop sending RTT information.</p>
Geographic location	Enabled.	You can disable this metric.
Connection load	Disabled.	<p>You can enable this metric.</p> <p>You also can change the data collection interval, the number of intervals used to calculate the connection load average, and the relative weights of the intervals.</p>

**Table 9.2: GSLB Policy Metrics (Continued)**

<b>metric</b>	<b>Default</b>	<b>Configuration Options</b>
Available session capacity	<p>Enabled.</p> <p>The default tolerance is 10%.</p> <p>When comparing sites based on the session table utilization, the GSLB ServerIron will prefer one site over the other only if the difference in session table utilization is greater than the tolerance percentage.</p>	<p>You can change the tolerance to a value from 0 – 100%.</p> <p>You also can disable this metric.</p>
FlashBack speed (how quickly the GSLB receives the Layer 4 TCP and Layer 7 health check results)	<p>Enabled.</p> <p>The default tolerance is 10%. This applies to the TCP health check and application health checks.</p> <p>When comparing sites based on the FlashBack speed, the GSLB ServerIron will prefer one site over the other only if the FlashBack speeds differ by more than the specified tolerance.</p>	<p>You can change the TCP and application tolerances individually. A change applies to all the TCP ports or applications at the remote site.</p> <p>You also can disable this metric.</p>
Administrative Preference	<p>Disabled.</p> <p>When enabled, the default preference is 128. The GSLB ServerIron will prefer the site with the highest administrative preference. If you set the preference for a site ServerIron to 0, the site is administratively removed from GSLB selection.</p>	<p>You can enable this metric. On an individual site ServerIron basis, you can change the preference from 128 (the default) to a value from 0 – 255.</p>
Least Response selection (the site ServerIron that has been selected less often than others)	Enabled.	Not configurable.
Round Robin Selection	<p>Disabled.</p> <p>When Round Robin Selection is enabled, Least Response Selection is disabled. Round Robin Selection is an alternative to Least Response Selection. They are mutually exclusive.</p> <p>Like Least Response Selection, Round Robin Selection is a tie breaker, used only if two or more sites are equal following comparison against all other enabled metrics.</p>	Not configurable.

To change the GSLB policy, use either of the following methods.

After changing policy values, you can display the new values using the **show gslb policy** command. If you decide you want to change a value back to its default (using “no” in front of the command you used to change it), you can

display all the default policy values by entering the **show gslb default** command. See “Displaying the GSLB Policy” on page 9-67.

---

**NOTE:** You also can configure the ServerIron to intercept or directly respond to DNS queries instead of evaluating responses from the authoritative DNS server. See “Configuring DNS Cache Proxy” on page 9-47 and “Configuring Transparent DNS Query Intercept” on page 9-50.

---

### Re-Ordering the GSLB Policy Metrics

You can change the order in which the GSLB ServerIron applies the policy metrics. To change the order, specify the metrics in the desired order.

---

**NOTE:** Foundry Networks recommends that you always use the health check as the first metric. Otherwise, it is possible that the GSLB policy will not select a “best” choice, and thus send the DNS reply unchanged. For example, if the first metric is geographic location, and the DNS reply contains two sites, one in North America and the other in South America, for clients in South America the GSLB policy favors the South American site after the first comparison. However, if that site is down, the GSLB policy will find that none of the sites in the reply is the “best” one, and thus send the reply unchanged.

You cannot change the position of the Least Response Selection or Round Robin Selection metric, whichever is enabled. The GSLB ServerIron uses the Least Response Selection or Round Robin Selection metric as a tie-breaker if the other comparisons do not result in selection of a “best” site.

---

### USING THE CLI

To specify a new GSLB policy order, enter a command such as the following:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# health-check metric-order set round-trip-time
capacity num-session flashback
```

This command changes the GSLB policy to the following:

- The health-check results
- The round-trip time between the remote ServerIron and the DNS client
- The site ServerIron’s session capacity threshold
- The site ServerIron’s available session capacity
- The site ServerIron’s FlashBack speed (how quickly the GSLB receives the health check results)
- The Least Response selection (the site ServerIron that has been selected less often than others)

Two of the metrics, server health and geographic location, are not specified. As a result, these metrics are not used when evaluating site IP addresses in the DNS responses.

To display the GSLB policy after you change it, enter the **show gslb policy** command. See “Displaying the User-Configured GSLB Policy” on page 9-70.

**Syntax:** [no] metric-order set <list>

The <list> parameter is a list of the metrics you want to use, in the order you want the GSLB ServerIron to use them. The GSLB uses the metrics in the order you specify them. You can specify one or more of the following:

- **capacity** – The site ServerIron’s available session capacity
- **connection-load** – The site ServerIron’s average number of new connections per second
- **flashback** – The site ServerIron’s FlashBack speed (how quickly the GSLB receives the health check results)
- **geographic** – The geographic location of the server

- **health-check** – The Layer 4 and application health checks
- **num-session** – The site ServerIron's session capacity threshold
- **preference** – The administratively configured preference for the site ServerIron
- **round-trip-time** – The round-trip time between the remote ServerIron and the DNS client

There are no parameters for the Least Response Selection or Round Robin Selection metrics. These metrics are tie-breakers. Only one of them is enabled at a time and the one that is enabled is always the last metric in the policy.

To reset the order of the GSLB policy metrics to the default (and also re-enable all disabled metrics), enter the following command:

```
ServerIron(config-gslb-policy)# metric-order default
```

**Syntax:** metric-order default

The **no metric-order set** command also resets the order and re-enables all disabled metrics. This command is equivalent to **metric-order default**.

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the Policy link to display the GSLB Policy panel, as shown in Figure 9.9 on page 9-27.
5. For each metric, select the new order position from the Order field's pulldown menu for that metric. For example, to change the order of the session capacity threshold and RTT metrics, select 3 from the Session Capacity Threshold field's pulldown menu and select 2 from the Round Trip Time field's pulldown menu. If you select 0, the metric is disabled. (This is the same as removing the checkmark from the metric's Enable checkbox.)
6. Click Apply to implement the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

#### Disabling or Enabling GSLB Policy Metrics

You can explicitly disable individual GSLB policy metrics.

---

**NOTE:** If you explicitly disable both the health check and flashback metrics, the GSLB ServerIron does not perform any health checks on the remote sites. If you disable the RTT metric, the GSLB ServerIron instructs the site ServerIron to stop sending RTT information.

---

#### USING THE CLI

To disable individual GSLB policy metrics, enter commands such as the following:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# no health-check
ServerIron(config-gslb-policy)# no geographic
```

The example above disables the health check and geographic metrics.

**Syntax:** [no] health-check | num-session | preference | round-robin | round-trip-time | geographic | connection-load limit <average-load> | capacity | flashback

To enable a metric, enter the command without **no** in front of it. For example, to re-enable both the metrics disabled in the example above, enter the following commands:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# health-check
```

```
ServerIron(config-gslb-policy)# geographic
```

To enable the administrative preference metric, which is disabled by default, enter the following commands:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# preference
```

To specify the site connection limit and enable the connection limit metric, enter commands such as the following:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# connection-load limit 500
```

This command sets the site connection limit to 500 connections. During site comparison, the GSLB policy discards sites that have an average load of new connections that is higher than the amount you specify. All other sites are passed to the next GSLB policy metric as potential candidates.

**Syntax:** [no] connection-load limit <average-load>

You can specify from 1 – as high a value as you need. There is no default. You must specify a connection limit to enable the connection limit metric.

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the Policy link to display the GSLB Policy panel, as shown in Figure 9.9 on page 9-27.
5. Select the Enable checkbox to remove the checkmark next to the metric you want to disable.
6. Click Apply to implement the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

#### Modifying Connection Load Parameters

---

**NOTE:** This section applies only to ServerIrons running software release 07.2.25 or later and 07.3.04 or later.

---

A GSLB site's **connection load** is the average number of new connections per second on the site, over a given number of intervals. When you enable this GSLB metric, all potential candidates are compared against a predefined load limit. All sites that have fewer average connections than the threshold are selected and passed to the next comparison metric.

The connection load metric is disabled by default but is enabled (added to the GSLB policy) when you configure the metric.

#### Connection Load Parameters

You can configure the following parameters:

- Site connection limit
- Sampling intervals and sample rate
- Interval weights
- Comparison order in the GSLB policy

#### Site Connection Limit

The site connection limit is the maximum number of new connections per second a site can have without being disqualified by the GSLB policy. During site comparison, when the GSLB policy is comparing otherwise equal sites based on the connection load metric, the policy disqualifies a site if its average number of new connections is higher than the specified connection limit.

The same connection limit applies to all sites. You can specify from 1 – as high a value as you need. There is no default. When you specify a value, the connection load metric is enabled (added to the GSLB policy).

This is the only parameter that you are required to set for the metric. The other parameters have default values.

### **Sampling Intervals and Sample Rate**

The sampling interval is the number of data samples the GSLB controller averages together to calculate a site's connection load. The sample rate is the number of seconds between intervals.

By default, each GSLB site takes five samples, at 5-second intervals. Using the default sampling interval and sample rate, the site takes samples after 5 seconds, 10 seconds, 15 seconds, 20 seconds, and 25 seconds.

The number of new connections the site has at each of the five intervals is averaged together. This average value is the one the GSLB controller uses for the comparison.

- You can specify from 1 – 8 sampling intervals. The default is 5.
- You can specify from 5 – 60 seconds for the sample rate. The default is 5 seconds.

At any given time, the average connection load for a site is the average of the latest full set of data samples. For example, if the sampling interval is 5, then the average load is the average of the five most recent samples.

---

**NOTE:** The accuracy of the average is affected by the initial sampling rate. For example, if the sampling rate is 5 seconds, the average at the seventh second will consist of the average for the first through fifth seconds, rather than an average for the second through seventh seconds.

---

### **Interval Weights**

The interval weights are the relative weights of each data sample within a set of sampling intervals. When the data samples are averaged together, the relative weights of the samples can affect the outcome. You can adjust the load calculation formula by changing the weights of the intervals, so that some intervals are counted more heavily towards the average than other intervals. You can even eliminate the effect of an interval by setting its weight to 0.

For example, if a sampling interval contains six data samples and you assign higher weights to the third and fourth samples than to the others, the third and fourth samples play a larger role when the average connection load is calculated.

The default weight for each interval is 1. You can individually change the weight to a value from 0 – 10. If you set an interval's weight to 0, that interval is not included when the intervals are averaged together.

### **Comparison Order**

When the connection load metric is enabled, by default the metric is used after the geographic location metric but before the session capacity metric. You can change the order in which the metrics are applied.

#### ***Configuring the Connection Limit Metric***

To configure the connection limit metric, perform the following tasks on the ServerIron that is the GSLB controller. You do not need to perform any tasks on the site ServerIrons. All configuration for the metric takes place on the controller.

- Specify the site connection limit. Specifying the site connection limit also enables the metric in the GSLB policy.
- Optional – Change the sampling intervals and sample rate.
- Optional – Change the relative weights of the sampling intervals.
- Optional – Change the position of the metric in the GSLB policy. By default, the metric comes after comparison of geographic locations and before comparison of session capacities.

### **Specifying the Site Connection Limit**

To specify the site connection limit and enable the connection limit metric, enter commands such as the following:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# connection-load limit 500
```



This command sets the site connection limit to 500 connections. During site comparison, the GSLB policy discards sites that have an average load of new connections that is higher than the amount you specify. All other sites are passed to the next GSLB policy metric as potential candidates.

**Syntax:** [no] connection-load limit <average-load>

You can specify from 1 – as high a value as you need. There is no default. You must specify a connection limit to enable the connection limit metric.

### Changing the Sampling Intervals and Sample Rate

By default, the site ServerIron samples the load of new connections every five seconds and stores the average connection load for five intervals: the average loads at the previous 5, 10, 15, 20, and 25 seconds.

You can change the sampling interval and sample rate. Enter a command such as the following at the GSLB policy level of the CLI:

```
ServerIron(config-gslb-policy)# connection-load intervals 6 5
```

This command changes the number of sampling intervals from 5 to 6 but leaves the sample rate set to 5 seconds. At any given time, the site ServerIron will have the average load for six intervals, for the previous 5, 10, 15, 20, 25, and 30 seconds. The average connection load will be calculated based on these six samples.

**Syntax:** [no] connection-load intervals <num-intervals> <sampling-rate>

The <num-intervals> parameter specifies the number of samples you want the site ServerIron to collect and average together. You can specify 1 – 8 intervals. The default is 5.

The <sampling-rate> parameter specifies the number of seconds between each sample. You can specify 1 – 60 seconds. The default is 5 seconds.

### Changing the Sample Weights

By default, the site ServerIron weighs each data sample equally when calculating the connection average for the GSLB policy. The weight of each interval is 1.

You can change the weights to give more emphasis to some intervals and less emphasis to others. For example, if you are using five intervals, all five have equal influence on the average load calculated by the GSLB policy. If you want to give more emphasis to the third interval, you can give the third interval a higher weight than the other intervals. To ignore an interval when calculating the average, assign the weight 0 (zero) to the interval.

To change sample weights, enter a command such as the following at the GSLB policy level of the CLI:

```
ServerIron(config-gslb-policy)# connection-load weights 1 1 3 1 1
```

This command gives more weight to the third sampling interval than to the other intervals, while including all intervals in the calculation of the average connection load.

**Syntax:** [no] connection-load weights <weight1> [<weight2>...<weight8>]

The <weight> parameters specify the weights. You can specify from 0 – 10. If you enter 0, the interval is not included when calculating the average load. Enter the weights in the same order as the sampling intervals.

You do not need to enter weight values for all the intervals once you enter the last non-zero weight. For example, if you want to set the weight for interval three to 1 but use 0 for the weights of all the other intervals, you can enter the following command:

```
ServerIron(config-gslb-policy)# connection-load weights 0 0 1
```

When this command is entered, the weights for the fourth interval and higher are set to 0.

### Modifying the Session-Table Capacity and Threshold Tolerance Values

You can change the following parameters associated with the session-table metrics:

- Session capacity threshold – Specifies how close to the maximum session capacity the site ServerIron (remote ServerIron) can be and still be eligible as the best site for the client. This mechanism provides a way to shift load away from a site before the site becomes congested. The default value for the threshold is 90%. Thus a site ServerIron is eligible to be the best site only if its session utilization is below 90%.

- Available session capacity tolerance – Specifies the percentage by which the number of available sessions on the site ServerIron can differ from the number of available sessions on another site ServerIron and still be considered an equally good site.

You can change these parameters on an individual basis. To do so, use either of the following methods.

#### *USING THE CLI*

To change the session-table capacity metric, enter commands such as the following:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy) # capacity threshold 99
```

**Syntax:** [no] capacity threshold <num>

The <num> parameter specifies the maximum percentage of a site ServerIron's session table that can be in use. If the ServerIron's session table utilization is greater than the specified percentage, the GSLB ServerIron prefers other sites over this site. You can specify a percentage from 0 – 100. The default is 90.

To change the session-table tolerance metric, enter commands such as the following:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy) # num-session tolerance 20
```

**Syntax:** [no] num-session tolerance <num>

The <num> parameter specifies the maximum percentage by which the session table utilization on ServerIrons at different sites can differ without the GSLB ServerIron selecting one over the other based on this metric. You can specify a tolerance from 0 – 100. The default is 10.

#### *USING THE WEB MANAGEMENT INTERFACE*

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the Policy link to display the GSLB Policy panel, as shown in Figure 9.9 on page 9-27.
5. Edit the value in the Threshold field under Session Capacity Threshold to change this parameter. You can specify a percentage from 0 – 100. The default is 90.
6. Edit the value in the Tolerance field under Available Session Capacity to change this parameter. You can specify a tolerance from 0 – 100. The default is 10.
7. Click Apply to implement the change.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

#### **Modifying the FlashBack Tolerance Values**

You can modify the following FlashBack parameters:

- Application tolerance
- TCP tolerance

The GSLB ServerIron uses a tolerance value when comparing the FlashBack speeds of different sites. The tolerance value specifies the percentage by which the FlashBack speeds of the two sites must differ in order for the ServerIron to choose one over the other. The default FlashBack tolerance is 10%. Thus, if the FlashBack speeds of two sites are within 10% of one another, the ServerIron considers the sites to be equal. However, if the speeds differ by more than 10%, the ServerIron prefers the site with the lower FlashBack speed.

FlashBack speeds are measured at Layer 4 for all TCP/UDP ports. For the application ports known to the ServerIron, the FlashBack speed of the application is also measured.

When the ServerIron compares the FlashBack speeds, it compares the Layer 7 (application-level) FlashBack speeds first, if applicable. If the application has a Layer 7 health check and if the FlashBack speeds are not equal,

the ServerIron is through comparing the FlashBack speeds. However, if only the Layer 4 health check applies to the application, or if further tie-breaking is needed, the ServerIron then compares the Layer 4 FlashBack speeds.

To modify the application (Layer 7) tolerance, TCP (Layer 4) tolerance, or both, use either of the following methods.

#### USING THE CLI

To change the tolerances for the response times of TCP and application health checks, when used as a metric for selecting a site, enter commands such as the following:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# flashback application tolerance 30
ServerIron(config-gslb-policy)# flashback tcp tolerance 50
```

**Syntax:** [no] flashback application | tcp tolerance <num>

The **application** | **tcp** parameter specifies whether you are modifying the tolerance for the Layer 4 TCP health check or the Layer 7 application health checks. You can change one or both and the values do not need to be the same. For each, you can specify from 0 – 100. The default for each is 10.

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the Policy link to display the GSLB Policy panel, as shown in Figure 9.9 on page 9-27.
5. Edit the value in the Application tolerance field under FlashBack to change this parameter. You can specify a tolerance from 0 – 100. The default is 10.
6. Edit the value in the TCP tolerance field under FlashBack to change this parameter. You can specify a tolerance from 0 – 100. The default is 10.
7. Click Apply to implement the change.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

#### Modifying Round-Trip Time Values

The Round-trip time (RTT) is the amount of time that passes between when the remote site receives a TCP connection (sends a TCP SYN) from the client and when the remote site receives the client's acknowledgment of the connection request (sends a TCP ACK). A site ServerIron sends RTT data to the GSLB ServerIron every one second (software release 07.1.x, 07.3.x, or 07.2.24 or earlier) or every five seconds (software release 07.2.25 or later).

You can modify RTT parameters to change processing of the RTT information reported by the GSLB and remote site ServerIrons. You can change the following parameters, on an individual basis:

- **RTT cache interval** – The site ServerIrons use the Foundry GSLB protocol to send RTT information to the GSLB ServerIron. The GSLB ServerIron stores this information in a cache. The GSLB ServerIron uses the entries in the cache when using the RTT metric to evaluate IP addresses in a DNS reply. Entries in the cache age out if they remain unused. The default aging interval for RTT cache entries is 120 seconds. You can change the interval to a value from 10 – 1,000,000 seconds (about 11-1/2 days).
- **RTT cache prefix** – The entries in the RTT cache include IP address information for the clients. To avoid overflowing the cache, cache entries are aggregated based on the IP information. For example, if the GSLB ServerIron receives RTT information for clients at 192.21.4.69 and 192.21.4.18, and the cache prefix is 31 bits, both addresses go in as separate entries. However, if the prefix is 16 bits, the GSLB ServerIron aggregates the addresses. In this case, only one entry, 192.21.x.x goes in the cache. The default number of bits in the prefix is 20. You can specify a value from 1 – 31.
- **RTT tolerance** – When the GSLB ServerIron compares two site IP addresses based on RTT, the GSLB ServerIron favors one site over the other only if the difference between the RTT values is greater than the

specified percentage. This percentage is the RTT tolerance. You can set the RTT tolerance to a value from 0 – 100. The default is 10%.

- RTT explore percentage – Site ServerIrons send RTT information only for the sessions that clients open with them. These are clients referred to the site ServerIron by the GSLB ServerIron. If the metrics that come before this one (based on the GSLB policy order) do not select a “best” site, the ServerIron selects a site based on RTT.

Since the only RTT information received by the GSLB ServerIron comes from the site ServerIrons to which the GSLB ServerIron has referred clients, it is possible for the GSLB ServerIron to continually bias its selection toward the first site ServerIron that sent RTT information. To prevent this from occurring, the GSLB ServerIron intentionally ignores the RTT metric for a specified percentage of the requests from a given client network. You can specify an RTT explore percentage from 0 – 100. The default is 5. By default, the GSLB ServerIron ignores the RTT for 5% of the client requests from a given network.

You also can add static RTT prefix cache entries.

#### *USING THE CLI*

Use the following procedures to change the RTT parameters using the CLI.

##### *Changing the RTT Cache Interval*

To change the RTT cache interval, enter commands such as the following:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# round-trip-time cache-interval 30
```

The command in this example changes the RTT cache interval from 10 seconds to 30 seconds.

**Syntax:** [no] round-trip-time cache-interval <num>

The <num> parameter specifies the aging interval and can be from 10 – 1,000,000 seconds (about 11-1/2 days). The default is 120 seconds.

##### *Changing the RTT Cache Prefix*

To change the RTT cache prefix, enter commands such as the following:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# round-trip-time cache-prefix 16
```

The command in this example changes the RTT cache prefix from 20 bits to 16 bits.

**Syntax:** [no] round-trip-time cache-prefix <num>

The <num> parameter specifies the number of significant bits in the prefix and can be from 1 – 32. The default is 20.

##### *Changing the RTT Tolerance*

To change the RTT tolerance, enter commands such as the following:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# round-trip-time tolerance 70
```

The command in this example changes the RTT tolerance from 10% to 70%.

**Syntax:** [no] round-trip-time tolerance <num>

The <num> parameter specifies the percentage above which the RTTs of two sites must differ for the GSLB ServerIron to favor one site over the other based on the RTT. You can specify a value from 0 – 100. The default is 10%.

##### *Changing the RTT Explore Percentage*

To change the RTT explore percentage, enter commands such as the following:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# round-trip-time explore-percentage 10
```

The command in this example changes the RTT explore percentage from 5% to 10%.

**Syntax:** [no] round-trip-time explore-percentage <num>

The <num> parameter specifies the explore percentage and can be from 0 – 100. The default is 5.

#### **USING THE WEB MANAGEMENT INTERFACE**

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the Policy link to display the GSLB Policy panel, as shown in Figure 9.9 on page 9-27.
5. Edit the value in the Cache Interval field under Round Trip Time to change this parameter. You can specify a value from 10 – 300 seconds. The default is 10 seconds.
6. Edit the value in the Explore Percentage field under Round Trip Time to change this parameter. You can specify a percentage value from 0 – 100. The default is 5.
7. Edit the value in the Prefix Length field under Round Trip Time to change this parameter. You can specify a value from 1 – 32. The default is 20.
8. Edit the value in the RTT Tolerance field under Round Trip Time to change this parameter. You can specify a percentage value from 0 – 100. The default is 10%.
9. Click Apply to implement the change.
10. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

#### **Configuring Static RTT Prefix Cache Entries**

The GSLB ServerIron maintains a cache of round-trip time (RTT) information received from the site ServerIrons through the GSLB protocol. The RTT is the amount of time that passes between when a remote site initiates a TCP connection from the client and when the remote site receives the client's acknowledgment of the connection request. Each site ServerIron sends RTT information to the GSLB ServerIron at one-second intervals.

The GSLB ServerIron uses the RTT information in the prefix cache when evaluating a site using the GSLB policy. Thus, the cache entry provides the RTT information used for the RTT metric during evaluation of the GSLB policy.

When the GSLB ServerIron receives RTT information from a site ServerIron, the IP address of the client is compared to the prefixes in the cache. If the address fits within a network in one of the prefixes, the GSLB ServerIron stores the RTT information for that site under the prefix entry. If the client address is within more than one prefix entry, the GSLB ServerIron selects the entry with the longer prefix (the more exact match).

The GSLB ServerIron makes a dynamic entry in the prefix cache of the length specified by the cache prefix the first time the ServerIron processes a DNS query or response from that prefix. After that, each time the GSLB ServerIron receives a subsequent DNS query from within that prefix, the ServerIron resets the aging timer for the cache prefix entry. If a dynamic entry is not refreshed by subsequent queries, the entry ages out.

You can manually add static prefix information to the cache. For example, you can add static cache entries with longer prefix information than the dynamic cache entries to ensure that RTT information is stored under the static entries instead of dynamic cache entries with shorter prefixes. This is useful when you want to ensure that certain prefixes are always present in the cache regardless of how often the GSLB ServerIron receives RTT data for them. Static prefixes do not age out.

---

**NOTE:** The GSLB ServerIron uses the most exact match when more than one prefix entry can apply to the same site address. To ensure that the GSLB ServerIron uses a static entry instead of certain dynamic entries for a given address, make sure prefix of the static entry is longer than the prefix for dynamic entries.

---

---

**NOTE:** Since RTT information is stored under individual domain names that are queried, the RTT information reported from remote ServerIrons are not recorded under the static records until the GSLB ServerIron receives the first DNS query or response.

---

To enter a static cache entry in the prefix cache, use the following CLI method.

### USING THE CLI

To add a static prefix cache entry, enter commands such as the following:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# static-prefix 61.1.1.1/20
```

**Syntax:** static-prefix <ip-addr>/<prefix-length>

The <ip-addr> specifies the address of the cache entry. This is not necessarily the address of a remote site. The address you specify here is combined with the prefix length to result in a network prefix (network portion of an IP address). The prefix length can be from 1 – 31.

---

**NOTE:** The prefix length 0 is not applicable to this feature and is ignored by the software.

---

You can enter more than one prefix on the same command line. Separate each prefix with a space. You can configure up to 250 static prefixes on a ServerIron.

The command in this example configures an entry for address 61.1.1.1 with a prefix of 20 bits. (Due to the prefix length, the value actually stored in the cache is 61.1.0.0.20.) When the GSLB ServerIron receives RTT information for an address within the specified prefix, the GSLB ServerIron stores the information in the static prefix entry configured above, instead of creating a dynamic entry.

### USING THE WEB MANAGEMENT INTERFACE

You cannot add a static cache entry using the Web management interface.

## Configuring Affinity

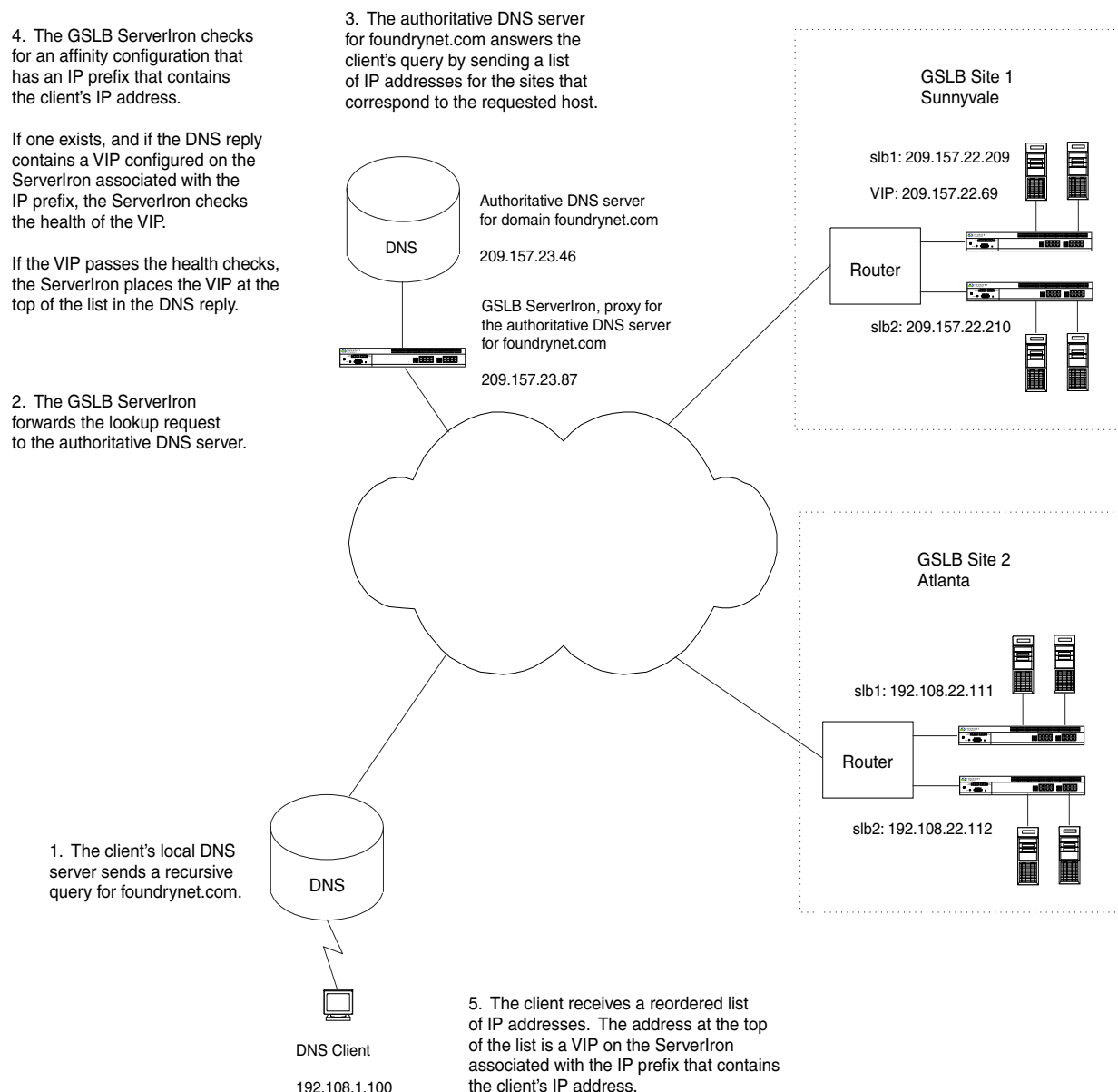
The Affinity feature configures the GSLB ServerIron to always prefer a specific site ServerIron for queries from clients whose addresses are within a given IP prefix. This feature is useful in the following situations:

- When you want to use a primary site for all queries and use other sites only as backups.
- When you want to use a site located near clients within a private network for all queries from the private network.

To configure affinity, you associate a site ServerIron with an IP prefix. When the GSLB ServerIron receives a query from a client whose IP address is within the configured prefix, the GSLB ServerIron examines the DNS reply for a virtual IP address (VIP) configured on the ServerIron associated with the IP prefix that contains the client's IP address.

Figure 9.10 shows an example of the affinity feature. In this example, the GSLB ServerIron contains the following affinity configuration:

- IP prefix: 192.0.0.0/8, Site ServerIron: 209.157.22.209 (slb1 in the sunnyvale site)

**Figure 9.10 Example of the affinity feature**

In Figure 9.10, the client's IP address is within the configured affinity prefix, so the ServerIron checks the DNS reply for a VIP configured on the ServerIron associated with the prefix.

- If the reply contains a VIP on the ServerIron associated with the prefix that the client's IP address is in, the ServerIron places the VIP at the top of the address list in the reply. (This assumes that the VIP passes the applicable health checks if they are enabled.)
- If the reply contains more than one VIP on the ServerIron associated with the prefix that contains the client's IP address, the ServerIron selected the VIP that has been selected least often. (This is the last metric in the GSLB policy and is used as a tiebreaker).
- If the VIP fails a health check, or if the reply does not contain a VIP on the ServerIron associated with the prefix that contains the client's IP address, the ServerIron uses the other GSLB metrics in the policy to reorder the list.



You can configure up to 50 affinities. The IP prefix in each affinity can have a value from 0 – 31. You can associate only one ServerIron with a prefix. However, you can associate multiple prefixes with the same ServerIron.

If you configure more than one affinity, it is possible for a client's IP address to be within the prefixes of more than one affinity definition. In this case, the ServerIron uses the affinity whose prefix is a more specific match for the client. For example, assume that the GSLB ServerIron in Figure 9.10 contains the following affinities:

- IP prefix: 192.0.0.0/8, Site ServerIron: 209.157.22.209 (slb1 in the sunnyvale site)
- IP prefix: 192.108.0.0/16, Site ServerIron: 209.157.22.210 (slb2 in the sunnyvale site)

The client IP address 192.108.1.100 falls within both prefixes. However, prefix 192.108.0.0/16 is a more precise match than prefix 192.0.0.0/8. Therefore, the ServerIron uses the affinity definition that contains prefix 192.108.0.0/16. If the VIP for the more precise prefix is not available (for example, if it fails a health check), the ServerIron uses the standard GSLB policy to select the best site.

You can configure a default affinity definition by using the prefix 0.0.0.0/0 in the definition. When you configure a default affinity definition, the ServerIron prefers a VIP on the ServerIron associated with the prefix 0.0.0.0/0 for all clients except those whose addresses are within a prefix configured in another affinity definition. Configuring a default affinity definition is optional. If you do not configure one, the ServerIron uses the standard GSLB policy for clients whose addresses are not within the prefix of an affinity definition.

### **Configuring an Affinity Definition**

To configure an affinity definition, use the following CLI method.

#### **USING THE CLI**

To configure an affinity definition, enter commands such as the following:

```
ServerIron(config)# gslb affinity
ServerIron(config-gslb-affinity)# prefer sunnyvale slb-1 for 0.0.0.0/0
ServerIron(config-gslb-affinity)# prefer atlanta slb-1 for 192.108.22.0/22
```

These commands configure a default affinity definition (using the 0.0.0.0/0) prefix and an affinity definition that uses prefix 192.108.22.0/22. For clients that are not within the prefix in the second affinity definition, the ServerIron uses the default affinity definition. The ServerIron sends clients whose IP addresses are within the 192.108.22.0/22 prefix to a VIP on slb-1 at the “atlanta” site, when available. The ServerIron sends all other clients to a VIP on slb-1 at the “sunnyvale” site when available.

#### **Syntax:** gslb affinity

This command places the CLI at the affinity configuration level.

**Syntax:** [no] prefer <site-name> <si-name> | <si-ip-addr> for <ip-addr> <ip-mask> | <ip-addr>/<prefix-length>

You can refer to the ServerIron by its GSLB site name and ServerIron name or by its management IP address. Use one of the following parameters:

- The <site-name> and <si-name> parameters specify the remote site and a ServerIron at that site. If you use this method, you must specify both parameters.
- The <si-ip-addr> parameter specifies the site ServerIron's management IP address.

---

**NOTE:** In either case, the running-config and the startup-config file refer to the ServerIron by its IP address.

---

The <ip-addr> <ip-mask> or <ip-addr>/<prefix-length> parameter specifies the prefix. You can specify a mask from 0.0.0.0 – 255.255.255.254. If you instead specify a prefix length, you can specify from 0 – 31 bits.

If you specify 0.0.0.0 0.0.0.0 or 0.0.0.0/0, the ServerIron applies the affinity definition to all client addresses. As a result, an address that does not match another affinity definition uses the zero affinity definition by default. If you do not configure a default affinity definition, the ServerIron uses the standard GSLB policy for clients whose addresses are not within a prefix in an affinity definition.

#### **USING THE WEB MANAGEMENT INTERFACE**

You cannot configure this parameter using the Web management interface.



### Displaying Affinity Configuration Information

To display affinity configuration information, use the following CLI method.

#### USING THE CLI

To display affinity configuration information, enter the following command at any level of the CLI:

```
ServerIron(config)# show gslb cache 192.108.22.0
prefix length = 22, prefix = 192.108.22.0, region = N-AM
prefix source = affinity, client-query
affinity = site: atlanta, SI: 192.108.22.111 slb-1

www.foundryet.com:
  site = atlanta, SI = slb-1(192.108.22.111), rtt = 4 (x100 usec)
```

**Syntax:** show gslb cache <ip-addr>

The <ip-addr> command specifies a site address.

The output in this example shows the information in the GSLB ServerIron's prefix cache for prefix 192.108.22.0, including the affinity configuration information.

The prefix source field indicates the source of the prefix. If the source is "affinity", the prefix is associated with a site ServerIron as part of an affinity definition.

If the ServerIron contains an affinity definition for the prefix you specify, the affinity information is listed in the affinity field. The affinity field indicates the GSLB site, management IP address, and GSLB name of the ServerIron associated with the prefix.

#### USING THE WEB MANAGEMENT INTERFACE

You cannot display this information using the Web management interface.

## Configuring DNS Cache Proxy

The DNS cache proxy feature allows the ServerIron to act as a proxy for a DNS server, by responding directly to the client queries without forwarding them to the DNS server. Just as in the regular GSLB mode, the ServerIron periodically queries the authoritative DNS server for IP addresses corresponding to the domains configured for GSLB and caches them. However, unlike regular GSLB, the ServerIron does not forward every client query to the authoritative DNS server, it responds directly to the client using the cached address list for the requested domain.

When you enable DNS cache proxy, the ServerIron applies the GSLB policy to the IP addresses it has cached for the requested domain, and responds to the client with the best address. The ServerIron refreshes the address cache by periodically querying the authoritative DNS server. The default update interval is 30 seconds and is configurable.

The DNS cache proxy feature is useful in network environments where the traffic between the ServerIron and the authoritative DNS server introduces noticeable latency in the response to client requests. For example, if the ServerIron and the authoritative DNS server are connected over the Internet, this feature can eliminate the delays caused by that connection.

---

**NOTE:** You can configure the GSLB ServerIron to use an alias instead of the configured domain name when querying the DNS server for IP addresses. This feature is useful when the DNS server contains a set of proxy server addresses under the domain's alias, but contains the original site address under the real domain name. See "Configuring DNS Domain Name Aliases" on page 9-24.

---

In configurations where the ServerIron and DNS server are collocated, the additional round trip time between the ServerIron and DNS server is usually negligible. However, if the ServerIron and DNS server are in different networks, the delay can become significant. In this case, the DNS cache proxy can help enhance performance by eliminating the exchange between the ServerIron and DNS server for responses to client queries.

The DNS cache proxy feature is disabled by default. When the feature is disabled, the ServerIron forwards client requests to the actual DNS server, applies the GSLB policy to the responses, then sends the optimized response to the client. In this case, the round trip time between the ServerIron and DNS server is part of the overall round trip time between when the client sends the request and when the client receives the response.

If the GSLB ServerIron cannot respond directly to the client for the requested domain (for example, because the domain is not configured on the GSLB ServerIron), the ServerIron sends the request through to the DNS server. This is the same behavior as when the DNS cache proxy feature is disabled.

---

**NOTE:** You can combine the DNS cache proxy feature with the DNS override feature (added in software release 06.0.03) to completely eliminate the separate DNS server. In this case, the ServerIron contains all the required DNS information. See “Combining the DNS Cache Proxy and DNS Override Features” on page 9-49.

---

## Enabling DNS Cache Proxy

To enable DNS cache proxy, use either of the following methods.

### USING THE CLI

To enable DNS cache proxy, enter the following commands:

```
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# dns cache-proxy
```

**Syntax:** [no] dns cache-proxy

### USING THE WEB MANAGEMENT INTERFACE

You cannot configure this parameter using the Web management interface.

## Displaying the DNS Cache Proxy State

To determine whether DNS cache proxy is enabled, display the current GSLB policy settings the following CLI method.

### USING THE CLI

To display the current GSLB policy settings, which include the DNS cache proxy state, enter the following command at any level of the CLI:

```
ServerIron(config)# show gslb policy
  Default metric order: ENABLE
  Metric processing order:
    1-Server health check
    2-Remote SI's session capacity threshold
    3-Round trip time between remote SI and client
    4-Geographic location
    5-Remote SI's available session capacity
    6-Server flashback speed
    7-Least response selection

  DNS active-only: DISABLE  DNS best-only: DISABLE  DNS override: DISABLE
  DNS cache-proxy: ENABLE  DNS transparent-intercept: DISABLE
  remaining rows omitted for brevity...
```

**Syntax:** show gslb policy

The command output shown in bold type in the example indicates the DNS cache proxy state. The state can be one of the following:

- DISABLE (the default)
- ENABLE

### USING THE WEB MANAGEMENT INTERFACE

You cannot display this information using the Web management interface.

## Displaying Statistics for Transparent DNS Query Intercept or DNS Cache Proxy

The GSLB ServerIron maintains statistics for the transparent DNS query intercept and DNS cache proxy features.

The following statistics are displayed for DNS cache proxy:

- Number of DNS queries the GSLB ServerIron has responded to using the DNS cache proxy feature instead of forwarding the queries to the DNS server. (See the Direct response field under “DNS cache proxy stat:” in the output.)

The following statistics are displayed for transparent DNS query intercept:

- Number of queries the ServerIron has redirected to a proxy DNS server or another ServerIron. (See the Redirect field under “DNS query intercept stat:” in the output.)
- Number of queries to which the ServerIron has directly responded using a transparent DNS query intercept IP address configured on the ServerIron itself. (See the Direct response field under “DNS query intercept stat:” in the output.)

### USING THE CLI

To display DNS cache proxy statistics, enter the following command at any level of the CLI:

```
ServerIron(config)# show gslb global-stat
DNS cache proxy stat:
Direct response      =          10

DNS query intercept stat:
Redirect             =          0  Direct response      =          0
```

**Syntax:** show gslb global-stat

The Direct response field, under “DNS cache proxy stat”, lists how many DNS queries the GSLB ServerIron has responded to using the DNS cache proxy feature instead of forwarding the queries to the DNS server. In this example, the GSLB ServerIron has responded directly to client queries ten times with the best site address among those cached on the ServerIron itself, instead of forwarding the request to the DNS server.

For information about the statistics in the DNS query intercept stat section, see “Displaying Transparent DNS Query Intercept Statistics” on page 9-55.

### USING THE WEB MANAGEMENT INTERFACE

You cannot display this information using the Web management interface.

## Combining the DNS Cache Proxy and DNS Override Features

When the DNS cache proxy feature is enabled, the GSLB ServerIron has to query the authoritative DNS server at regular intervals, to refresh the IP address list for each domain configured for GSLB. You can eliminate the need for a backend DNS server, by combining the cache proxy feature with the DNS override feature.

When you enable the DNS override feature, you also need to configure an IP list for the required domains. The ServerIron performs health checks on the IP addresses configured for the domains and directly responds to client queries by using the GSLB policy to select the best IP address from the IP list configured for the requested domain.

By combining the DNS cache proxy feature with the DNS override feature, you can configure the ServerIron to directly respond to client requests, without ever consulting the authoritative DNS server.

**NOTE:** A GSLB ServerIron does not contain all the features of a real DNS server and thus cannot completely replace the DNS server.

**NOTE:** Although you do not need a real DNS server when you combine DNS cache proxy with DNS override, you still need to configure a virtual IP address for the DNS server. Clients send queries to the virtual IP address.

For information about configuring DNS cache proxy, see “Configuring DNS Cache Proxy” on page 9-47. For information about configuring DNS override, see “Enabling DNS Override” in the “Configuring Global Server Load Balancing” chapter of the April or later edition of the *Foundry ServerIron Installation and Configuration Guide*.

To add the virtual IP address for the DNS server, use the following CLI method.

#### USING THE CLI

To add a virtual IP address to which the clients can send their DNS queries, enter a command such as the following:

```
ServerIron(config)# server virtual-name dns-proxy 209.157.23.87
ServerIron(config-vs-dns-proxy)# port dns
```

The command in this example adds IP address 209.157.23.87 as a virtual server. When clients send their DNS queries to this address, the ServerIron processes the queries.

#### USING THE WEB MANAGEMENT INTERFACE

You cannot configure this parameter using the Web management interface.

## Configuring Transparent DNS Query Intercept

Transparent DNS query intercept allows a ServerIron to transparently intercept the DNS queries to the authoritative DNS server and redirect them to alternate DNS servers or handle them directly. This feature allows you to leave the IP address of the authoritative DNS server on the DNS itself server. You do not need to change the DNS server IP address as you do in standard GSLB configurations.

This feature is useful when you want to redirect clients for certain domains to proxy web servers, but you do not want to configure the proxy addresses on the DNS server itself or otherwise change the configuration of the DNS server.

**NOTE:** The ServerIron must be in the direct data path from all potential clients to the authoritative DNS server. Otherwise, it is possible for the DNS server to receive the queries instead of the ServerIron.

You can configure the following types of transparent DNS query intercept:

- Redirect the client queries to a proxy DNS server and perform GSLB on the response. The ServerIron redirects the client request to the alternate DNS server, applies the GSLB policy on the response and gives the best address(es) to the client.
- Redirect the client queries to a proxy DNS server and send the reply unchanged. The ServerIron redirects the client request to the alternate DNS server and sends the response, as is, to the client. The alternate DNS server could be a ServerIron configured for GSLB, in which case the reply has the best address(es) for the client.
- Directly respond to client queries using the IP addresses configured for the domain. The ServerIron does not forward or redirect the query to the actual or proxy DNS servers. Instead, it directly responds to the client by applying GSLB policy to pick the best IP address from among the IP list configured for the domain.

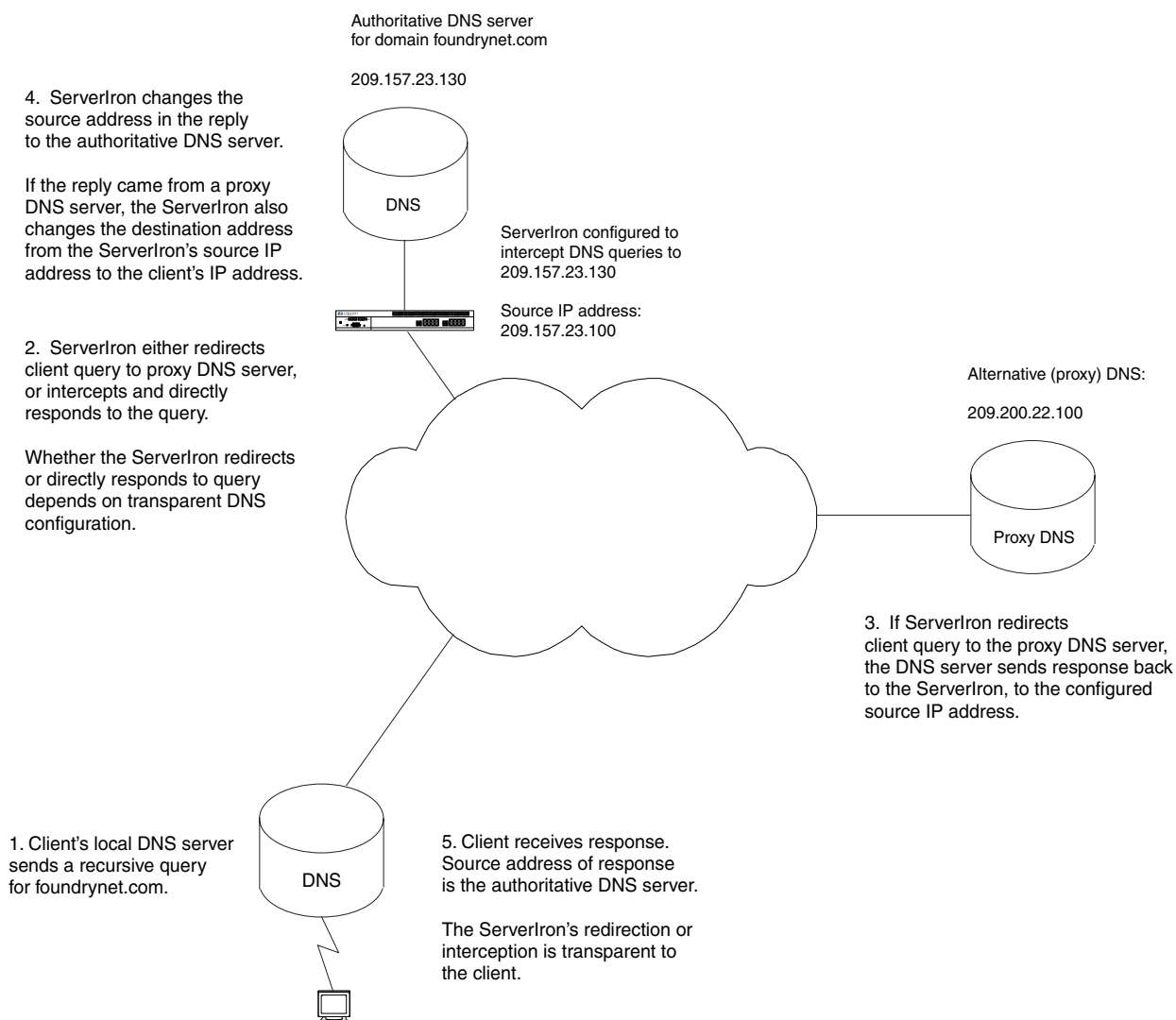
**NOTE:** The ServerIron configured for transparent intercept will redirect or directly respond to client requests only for domains configured on the ServerIron. If the domain name requested by the client is not configured on the ServerIron, it forwards the query to the actual DNS server without intercepting, and the reply is untouched by GSLB.

Figure 9.11 shows an example of a configuration that uses transparent DNS query intercept. In this example, the ServerIron is configured to intercept all client queries to the zone foundrynet.com and redirect them to the proxy DNS server and apply GSLB on the reply. The ServerIron uses its configured source-ip to make sure the DNS

reply from the proxy server comes to it, so that it can perform GSLB on the reply and send the best IP address(es) back to the client.

- The client's local DNS server sends a recursive query for foundrynet.com to the authoritative DNS server (209.157.23.130).
- The ServerIron intercepts and redirects client query to proxy DNS server (209.200.22.100).
- The proxy DNS server sends response back to the ServerIron's source IP address (209.157.23.100).
- The ServerIron changes the source address in the reply to the authoritative DNS server's address and the destination address from the ServerIron's source-IP to the client's IP address.
- The client receives the DNS response with the authoritative DNS server's source IP address. The ServerIron's interception and redirection is transparent to the client.

**Figure 9.11 Transparent DNS query intercept configuration**



### **Configuring Transparent DNS Query Intercept to Redirect Queries**

To configure transparent DNS query intercept to redirect queries to a proxy DNS server or another GSLB ServerIron:

- Configure a real server with the IP address of the proxy DNS server or other GSLB ServerIron to which you want to redirect queries.

- Configure a virtual server with the IP address of the authoritative DNS server that you want to intercept.
- Specify the domain and host application for which you want to intercept queries.
- Configure an IP policy to enable the ServerIron to examine incoming DNS packets.

---

**NOTE:** In standard GSLB configuration, the ServerIron sends a DNS query to the DNS server to get the IP addresses for the domain and performs health-checks on them. However in this transparent intercept mode, where you don't do GSLB on the DNS response, the ServerIron does not do these things.

---



---

**NOTE:** The ServerIron intercepts queries only for domain names configured on the ServerIron. For domain names that are not configured on the ServerIron, the ServerIron still sends queries to the authoritative DNS server.

---

Use the following CLI method to configure this feature.

#### *USING THE CLI*

To configure the ServerIron to redirect queries to an alternative DNS server, enter commands such as the following:

```
ServerIron(config)# source-ip 209.157.23.100 255.255.255.0 0.0.0.0
ServerIron(config)# server remote-name dns-redirect 209.200.22.100
ServerIron(config-rs-dns-redirect)# source-nat
ServerIron(config-rs-dns-redirect)# port dns
ServerIron(config-rs-dns-redirect)# exit
ServerIron(config)# server virtual-name dns-intercept 209.157.23.130 intercept
ServerIron(config-vs-dns-intercept)# port dns
ServerIron(config-vs-dns-intercept)# bind dns dns-redirect dns
ServerIron(config-vs-dns-intercept)# exit
ServerIron(config)# gslb dns zone foundrynet.com
ServerIron(config-gslb-dns-foundrynet.com)# host-info www http
ServerIron(config-gslb-dns-foundrynet.com)# exit
ServerIron(config)# ip policy 1 cache udp dns global
```

**Syntax:** [no] server source-ip <ip-addr> <ip-mask> <default-gateway>

---

**NOTE:** The gateway parameter is required. If you do not want to specify a gateway, enter "0.0.0.0".

---

This command adds a source IP address. The ServerIron uses the source IP address in packets that it sends to the alternative DNS server (the "real server"). Add an address that is in the same sub-net as the ServerIron's management IP address. If you do not add a source IP address and enable source NAT, the ServerIron leaves the client's IP address in the source address field of the redirected IP packets and as a result does not receive the alternative DNS server's responses. The ServerIron needs to receive the responses so it can modify the source IP address to match the address of the authoritative DNS server, so that when the client receives the response, the response appears to be from the authoritative DNS server. The redirection is thus transparent to the client.

**Syntax:** [no] server remote-name <name> <ip-addr>

This command adds the alternative DNS server (the one to which you want to redirect queries). You can enter this command multiple times for multiple alternative DNS servers.

---

**NOTE:** You can configure the alternate DNS server as a real server if it is in the same subnet as the ServerIron.

---

**Syntax:** [no] source-nat

This command enables source NAT. Source NAT allows the ServerIron to change the source IP address in the client request to one of the source addresses configured on the ServerIron. You must configure a source IP address and enable source NAT. You can enable source NAT globally or on individual real servers (as in the example above).

**Syntax:** [no] port dns

This command enables the DNS port on the real server. You must use this command so that the ServerIron knows you want to redirect DNS traffic to the real server (the alternative DNS server).

**Syntax:** [no] server virtual-name <name> <ip-addr> intercept

This command configures a virtual server that has the DNS server's actual IP address. When the ServerIron receives a DNS query addressed to the DNS server IP address, the ServerIron intercepts the packet instead of forwarding it to the DNS server. The **intercept** parameter is required and indicates that you want to use the virtual server for intercepting DNS queries. This parameter also instructs the ServerIron to ignore ARP requests and pings to the address. The ServerIron needs to ignore ARPs and pings to the address because the address still belongs to the authoritative DNS server itself. Without the **intercept** parameter, the ServerIron will respond to ARPs and pings to the virtual server's IP address.

**Syntax:** [no] bind dns <real-server-name> dns

This command binds the real server (the alternative DNS server) to the virtual server (the intercepted authoritative DNS server). This command creates an entry in the ServerIron's port binding table that allows the ServerIron to redirect DNS traffic sent to the authoritative DNS server to the alternative DNS server.

**Syntax:** [no] gslb dns zone-name <name>

This command specifies the zone for which you want to intercept queries. The ServerIron will intercept and redirect DNS queries only for the zones you specify, and forwards all other client queries to the authoritative DNS server.

**Syntax:** [no] host-info <host-name> <host-application> | <tcp/udp-portnum>

This command specifies the host application on the zone you specified above.

**Syntax:** ip policy <index> cache udp dns global

This command enables the ServerIron to examine incoming DNS packets. This command is required.

#### [USING THE WEB MANAGEMENT INTERFACE](#)

You cannot configure this feature using the Web management interface.

#### [Configuring Transparent DNS Query Intercept to Redirect Queries and Perform GSLB](#)

To configure transparent DNS query intercept to redirect queries to a proxy DNS server and perform GSLB on the response:

- Configure a real server with the IP address of the proxy DNS server
- Configure a virtual server with the IP address of the authoritative DNS server, which you want to intercept.
- Specify the domain and host application for which you want to intercept queries.
- Configure an IP policy to enable the ServerIron to examine incoming DNS packets.
- Enable port dns proxy on the real server corresponding to the proxy server.

---

**NOTE:** The ServerIron intercepts queries only for domain names configured on the ServerIron. For domain names that are not configured on the ServerIron, the ServerIron still sends queries to the authoritative DNS server.

---

To configure the ServerIron to redirect queries to another DNS server and apply GSLB on the response, enter commands such as the following:

```
ServerIron(config)# source-ip 209.157.23.100 255.255.255.0 0.0.0.0
ServerIron(config)# server remote-name dns-redirect 209.200.22.100
ServerIron(config-rs-dns-redirect)# source-nat
ServerIron(config-rs-dns-redirect)# port dns proxy
ServerIron(config-rs-dns-redirect)# exit
ServerIron(config)# server virtual-name dns-intercept 209.157.23.130 intercept
ServerIron(config-vs-dns-intercept)# port dns
```



```
ServerIron(config-vs-dns-intercept)# bind dns dns-redirect dns
ServerIron(config-vs-dns-intercept)# exit
ServerIron(config)# gslb dns zone foundrynet.com
ServerIron(config-gslb-dns-foundrynet.com)# host-info www http
ServerIron(config-gslb-dns-foundrynet.com)# exit
ServerIron(config)# ip policy 1 cache udp dns global
```

The commands are the same as the ones for configuring the ServerIron to redirect queries directly to another DNS server, with one difference. The command that enables the DNS port on the real server (the other ServerIron) uses the **proxy** parameter. This parameter indicates that the ServerIron needs to perform GSLB on the response before sending the response back to the client.

#### *USING THE WEB MANAGEMENT INTERFACE*

You cannot configure this feature using the Web management interface.

#### *Configuring Transparent DNS Query Intercept to Directly Respond to Queries*

To configure transparent DNS query intercept to directly respond to queries using IP addresses configured on the ServerIron:

- Configure a virtual server with the IP address of the authoritative DNS server that you want to intercept.
- Specify the domain name and host application for which you want to intercept queries.
- Enable the DNS transparent intercept feature.
- Configure an IP policy to examine incoming DNS packets.
- Enable **dns transparent-intercept** in the GSLB policy.

---

**NOTE:** In the direct-response mode, the ServerIron uses GSLB to pick the best address by default. No additional configuration is needed to further enable GSLB.

---

---

**NOTE:** The ServerIron intercepts queries only for domain names configured on the ServerIron. For domain names that are not configured on the ServerIron, the ServerIron still sends queries to the authoritative DNS server.

---

Use the following CLI method to configure this feature.

#### *USING THE CLI*

To configure the ServerIron to respond to queries using a set of IP addresses configured on the ServerIron itself, enter commands such as the following:

```
ServerIron(config)# server virtual-name dns-intercept 209.157.23.130 intercept
ServerIron(config)# gslb dns zone foundrynet.com
ServerIron(config-gslb-dns-foundrynet.com)# host-info www http
ServerIron(config-gslb-dns-foundrynet.com)# host-info www ip-list 209.200.1.1
209.200.1.2 209.200.1.3 209.200.1.4 209.200.1.5
ServerIron(config-gslb-dns-foundrynet.com)# exit
ServerIron(config)# gslb policy
ServerIron(config-gslb-policy)# dns transparent-intercept
ServerIron(config)# ip policy 1 cache udp dns global
```

These commands configure a virtual server for the authoritative DNS server IP address, specify the zone and host names for which to intercept queries, and specify the IP addresses to use in responses to the queries. The commands also enable the DNS transparent intercept feature and enable the ServerIron to examine incoming DNS packets.

Notice that unlike the types of transparent DNS query intercept shown in “Configuring Transparent DNS Query Intercept to Redirect Queries” on page 9-51, the type shown here does not require configuration of a real server. Since the ServerIron in this case is responding directly to the query instead of redirecting the query to another device, only the virtual server for intercepting the queries is required. Moreover, since the ServerIron is not redirecting the queries, you do not need to configure a source IP address and enable source NAT.



**Syntax:** host <host-name> ip-list <ip-addr...>

This command specifies the IP addresses you want the ServerIron to use in its replies to the intercepted DNS queries. You can specify as many addresses as you need. Separate each address with a space.

The ServerIron applies the GSLB policy to the addresses and sends only the best address in the response to a client query. If the GSLB policy does not result in a best address to send to the client, the ServerIron forwards the request to the authoritative DNS server. In either case, the source IP address in the response is the DNS server IP address, so the client always receives a response that appears to be from the DNS server.

**Syntax:** dns transparent-intercept

This command enables the DNS transparent intercept feature. You need to use this command only when you are configuring the type of transparent DNS query intercept that responds directly to the client. If you are configuring the type of transparent DNS query intercept that redirects the query to an alternative DNS server or to another ServerIron, do not use this command.

For information about the other commands, see “Configuring Transparent DNS Query Intercept to Redirect Queries” on page 9-51.

### USING THE WEB MANAGEMENT INTERFACE

You cannot configure this feature using the Web management interface.

### Displaying Transparent DNS Query Intercept Statistics

To display statistics for transparent DNS query intercept, use the following CLI method.

### USING THE CLI

To display transparent DNS query intercept statistics, enter the following command at any level of the CLI:

```
ServerIron(config)# show gslb global-stat
DNS cache proxy stat:
Direct response      =          0

DNS query intercept stat:
Redirect              =          10 Direct response      =          0
```

**Syntax:** show gslb global-stat

The transparent DNS query intercept statistics are displayed in the DNS query intercept stat section.

**Table 9.3: Transparent DNS Query Intercept Statistics**

This Field...	Displays...
Redirect	The number of queries the ServerIron has redirected to an alternative (proxy) DNS server or another ServerIron.
Direct response	The number of queries to which the ServerIron has directly responded using an IP address configured for the domain.

For information about the statistics in the DNS cache proxy stat section, see “Displaying Statistics for Transparent DNS Query Intercept or DNS Cache Proxy” on page 9-49.

### USING THE WEB MANAGEMENT INTERFACE

You cannot display this information using the Web management interface.

## Displaying GSLB Information

You display the following GSLB information:

- Site information – see “Displaying Site Information”
- Real server information on a remote ServerIron – see “Displaying Server Information for a Remote ServerIron” on page 9-61
- Zone and application information – see “Displaying Zone and Host Name Information” on page 9-62
- The default GSLB policy and the user-configured changes to the policy, if applicable – see “Displaying the GSLB Policy” on page 9-67
- RTT cache information for a specific IP address – see “Displaying RTT Prefix Cache Entries” on page 9-71
- GSLB resource usage and system capacity – see “Displaying GSLB Resource Utilization” on page 9-73
- Dynamic configuration information – see “Displaying Dynamic Configuration Information” on page 9-75

To display this information, use the methods described in the following sections.

## Displaying Site Information

You can display the following site information:

- ServerIron name and management IP address
- Site name (displayed only if you display information for all sites rather than an individual site)
- State of the GSLB protocol connection between GSLB ServerIron and site ServerIron
- Number of sessions in the ServerIron’s session table
- The percentage of the total number of sessions the ServerIron can maintain that are in use
- The percentage of the ServerIron’s CPU that is actively engaged in SLB and other activities
- The numeric preference value for this site ServerIron
- The geographic location of the ServerIron
- The virtual IP addresses (VIPs) configured on the ServerIron

To display information for the sites you have configured on the GSLB ServerIron, use either of the following methods.

*USING THE CLI*

To display information for all the configured sites, enter the following command at any level of the CLI:

```
ServerIron(config)# show gslb site
SITE: sunnyvale
SI: slb-1 209.157.22.209:
state: CONNECTION ESTABLISHED

Current num.  Session  CPU load  Preference  Location
sessions      util(%)   (%)
      500000      50        35   128        N-AM

Virtual IPs:
  209.157.22.227(A)        209.157.22.103(A)

SI: slb-2 209.157.22.210:
state: CONNECTION ESTABLISHED

Current num.  Session  CPU load  Preference  Location
sessions      util(%)   (%)
      1         0        16   128        N-AM

Virtual IPs:
  209.157.22.227(S)

SITE: atlanta
SI: slb-1 192.108.22.111:
state: CONNECTION ESTABLISHED

Current num.  Session  CPU load  Preference  Location
sessions      util(%)   (%)
      750000      75        41   128        N-AM

Virtual IPs:
  209.157.22.227(A)        209.157.22.104(A)

SI: slb-1 192.108.22.111:
state: CONNECTION ESTABLISHED

Current num.  Session  CPU load  Preference  Location
sessions      util(%)   (%)
      1         0        16   128        N-AM

Virtual IPs:
  209.157.22.227(S)
```

The following example shows information displayed when the connection-load metric is enabled.

```
ServerIron(config-gslb-policy)# show gslb site

SITE: two
SI: 1.1.1.2:
state: CONNECTION ESTABLISHED

Current num. Session CPU load Preference Location Connection
sessions      util(%)  (%)      (0-255)      N-AM      Load-Avg
           6         0         19         128      N-AM      30

Virtual IPs:
    1.1.1.12 (A)

Connection Load (Seconds:AvgLoad):
    5:36 10:34 15:32 20:31 25:30 30:28
```

**Syntax:** show gslb site [<name>]

The <name> parameter specifies a site name.

To display information about the GSLB site called “sunnyvale” and the ServerIrons providing SLB within those sites, enter the following command:

```
ServerIron(config)# show gslb site sunnyvale
SI: slb-1 209.157.22.209:
state: CONNECTION ESTABLISHED

Current num. Session CPU load Preference Location
sessions      util(%)  (%)      (0-255)      N-AM
           500000      50         35 128      N-AM

Virtual IPs:
    209.157.22.227 (A)

SI: slb-2 209.157.22.210:
state: CONNECTION ESTABLISHED

Current num. Session CPU load Preference Location
sessions      util(%)  (%)      (0-255)      N-AM
           1         0         16 128      N-AM

Virtual IPs:
    209.157.22.227 (S)
```

The **show gslb site** display shows the following information.

**Table 9.4: Global SLB Site Information**

This Field...	Displays...
ServerIron name and IP address	For each ServerIron, the first item of information listed is the name and management IP address. This is the information you specified when you added the ServerIron to the site.

Table 9.4: Global SLB Site Information (Continued)

This Field...	Displays...
SITE	<p>Indicates the site name of the ServerIron.</p> <p><b>Note:</b> This field appears only when you enter the <b>show gslb site</b> command without specifying a site name.</p>
SI	Indicates the site ServerIron name and management IP address.
State	<p>The state of the GSLB protocol connection between the GSLB ServerIron and the site ServerIron. The state can be one of the following:</p> <ul style="list-style-type: none"> <li>• <b>ATTEMPTING CONNECTION</b> – The GSLB ServerIron is still trying to establish a GSLB connection with the site ServerIron.</li> <li>• <b>CONNECTION ESTABLISHED</b> – The GSLB ServerIron has established a GSLB connection with the site ServerIron.</li> <li>• <b>SELF</b> – The GSLB ServerIron is also this site ServerIron.</li> </ul>
Current num. sessions	<p>The number of sessions in the ServerIron's session table. A session is a one-way connection to or from a real server.</p> <p>This information is reported by the site ServerIron.</p> <p><b>Note:</b> The number of sessions in the table does not necessarily match the number of active sessions on the real servers. This can occur if the session table contains sessions that are no longer active but have not yet timed out. See "Modifying Maximum Session Limit" on page 12-58 for information.</p>
Session util (%)	<p>The percentage of available sessions that are in use. This is the percentage of the total number of sessions the ServerIron can maintain that are in use. For example, if the ServerIron can maintain 1 million sessions (the default session capacity) and the session table contains 500,000 session entries, the session utilization is 50%.</p> <p>This information is reported by the site ServerIron.</p>
CPU load (%)	<p>The percentage of the ServerIron's CPU that is actively engaged in SLB and other activities.</p> <p>This information is reported by the site ServerIron.</p>
Preference	<p>The numeric preference value for this site ServerIron. The preference can be used by the GSLB policy to select a site. See "Site ServerIron's Administrative Preference" on page 9-8.</p> <p>This information is configured on the GSLB ServerIron.</p>

**Table 9.4: Global SLB Site Information (Continued)**

This Field...	Displays...
Location	<p>The geographic location of the ServerIron. The location is based on the ServerIron's management IP address and can be one of the following:</p> <ul style="list-style-type: none"> <li>• ASIA</li> <li>• EUROPE</li> <li>• N-AM – North America</li> <li>• S-AM – South America</li> </ul> <p><b>Note:</b> If you explicitly identified the geographic location, the value you specified appears instead of a value based on the IP address. See "Specifying the GSLB Sites and the Site ServerIrons" on page 9-19.</p>
Virtual IPs	<p>The virtual IP addresses (VIPs) configured on the ServerIron.</p> <p>This information is reported by the site ServerIron.</p> <p>The letter in parentheses at the end of each address indicates whether the ServerIron is an active or standby ServerIron for that address. The letter can be A (active) or S (standby). Unless the ServerIron is configured along with a partner ServerIron for Symmetric Server Load Balancing, the value is always A.</p> <p>If a number appears following the A or S, a host range (the unlimited VIP feature) is configured on the VIP. The number indicates the number of hosts in the host range.</p> <p><b>Note:</b> The GSLB ServerIron does not necessarily provide global SLB for all the VIPs configured on the site ServerIrons. The GSLB provides global SLB only for the VIPs that correspond to the DNS zone names you configure the GSLB ServerIron to load balance.</p>
Connection Load	<p>The average load at each connection-load sampling interval in the most recent set of sample intervals.</p> <p>In the example above, the connection load metric is configured to use six samples, at 5-second intervals. The sampling intervals and the average new-connection load at each interval are shown. On this site ServerIron, the average new-connection load for the last five seconds is 36, the average new-connection load for the last 10 seconds is 34, the average new-connection load for the last 15 seconds is 32, the average new-connection load for the last 20 seconds is 31, and so on. Any time you enter the command for this site ServerIron, the average load for the last 30 seconds is shown.</p> <p><b>Note:</b> This field applies only to ServerIrons running software release 07.2.25 or later and 07.3.04 or later.</p>

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.

4. Select the [Site](#) link to display the GSLB site information. See the previous section for descriptions of the information fields.

## Displaying Server Information for a Remote ServerIron

Generally, remote ServerIrons in a GSLB configuration are themselves configured with real servers and virtual servers. The real servers are the actual file servers for which the remote ServerIron provides load balancing. The virtual servers are the logical IP addresses that are published instead of the real server IP addresses.

The GSLB protocol allows you to query the site ServerIrons for configuration information as well as the session and CPU information used by the GSLB policy. You can view detailed configuration information and statistics for the site ServerIron, from the GSLB management console. You can display the following information:

- Real server configuration
- Virtual server configuration
- Port binding information (for the bindings between TCP/UDP ports on the real servers and the virtual server that represents the real servers)
- Session statistics for sessions between clients and the real servers

To display this information from the GSLB ServerIron's CLI, use either of the following methods.

### USING THE CLI

To display real server information for the real servers configured on a remote ServerIron, enter commands such as the following at any level of the GSLB ServerIron's CLI:

```
ServerIron(config)# rshow 209.157.22.209 server real
Real Servers Info

Name : rs1                               Mac-addr: abcd.5a11.d042
IP:10.10.10.1      Range:1      State:Active      Wt:1      Max-conn:1000000

Port      State      Ms  CurConn  TotConn  Rx-pkts  Tx-pkts  Rx-octet  Tx-octet  Reas
----      -
ftp       enabled  0  0        0        0        0        0        0        0
http      enabled  0  0        0        0        0        0        0        0
default   unbnd   0  0        0        0        0        0        0        0
Server    Total      0  0        0        0        0        0        0        0

Name : rs2                               Mac-addr: abcd.5a11.d043
IP:10.10.10.2      Range:1      State:Active      Wt:1      Max-conn:1000000

Port      State      Ms  CurConn  TotConn  Rx-pkts  Tx-pkts  Rx-octet  Tx-octet  Reas
----      -
ftp       enabled  0  0        0        0        0        0        0        0
http      enabled  0  0        0        0        0        0        0        0
default   unbnd   0  0        0        0        0        0        0        0
Server    Total      0  0        0        0        0        0        0        0
```

The command in this example displays real server configuration information for the remote ServerIron with management IP address 209.157.22.209. As shown in Figure 9.1 on page 9-3, this ServerIron is part of the “sunnyvale” site and is configured to load balance two real servers. In this example, the real servers are named rs1 and rs2. For descriptions of the information shown by this command, see “Displaying Real Server Information” on page 6-74.

**Syntax:** rshow <remote-ip-addr> server real | virtual | session | bind

The <remote-ip-addr> parameter specifies the remote ServerIron's management IP address.

The **real | virtual | session | bind** parameter specifies the information you want to display:

- **real** – displays real server information. This option is equivalent to entering the **show server real** command on the remote ServerIron. See “Displaying Real Server Information” on page 6-74.
- **virtual** – displays virtual server information. This option is equivalent to entering the **show server virtual** command on the remote ServerIron. See “Displaying Virtual Server Information” on page 6-82.
- **session** – displays session statistics. This option is equivalent to entering the **show server session** command on the remote ServerIron. See “Displaying Session Statistics” on page 6-93.
- **bind** – displays port binding information. This option is equivalent to entering the **show server bind** command on the remote ServerIron. See “Displaying Port-Binding Information” on page 6-92.

#### USING THE WEB MANAGEMENT INTERFACE

You cannot use the Web management interface to display remote ServerIron real and virtual server information.

## Displaying Zone and Host Name Information

To display information about the DNS zones and host names that you have configured the GSLB ServerIron to globally load balance, use either of the following methods.

---

**NOTE:** If you also want to display information about the site and ServerIron on which a VIP is configured, use the **show gslb dns detail** command instead. See “Displaying Detailed DNS Zone Information” on page 9-65.

---

#### USING THE CLI

To display information about all the DNS zones and host applications configured on the GSLB ServerIron, enter the following command at any level of the CLI:

```
ServerIron(config)# show gslb dns zone
ZONE: foundrynet.com
HOST: www:
```

					Flashback delay (x100us)		DNS resp. selection percentage (%)
					TCP	APP	
* 209.157.22.227:	dns	v-ip	ACTIVE	N-AM.	6	60	40
* 209.157.22.228:	dns	v-ip	ACTIVE	N-AM.	3	30	60
* 210.224.100.5:	dns	real-ip	DOWN	ASIA	--	--	0
* 201.100.100.6:	dns	real-ip	DOWN	S-AM.	--	--	0
* 213.34.100.4:	dns	real-ip	DOWN	EUROPE	--	--	0

```
HOST: ftp:
```

					Flashback delay (x100us)		DNS resp. selection percentage (%)
					TCP	APP	
* 209.157.22.103:	dns	v-ip	ACTIVE	N-AM.	6	60	40
* 209.157.22.104:	dns	v-ip	ACTIVE	N-AM.	3	30	60
* 210.224.100.7:	dns	real-ip	DOWN	ASIA	--	--	0
* 201.100.100.8:	dns	real-ip	DOWN	S-AM.	--	--	0
* 213.34.100.9:	dns	real-ip	DOWN	EUROPE	--	--	0

**Syntax:** show gslb dns zone [<name>]

The <name> parameter specifies the zone name.

To display GSLB information for a specific DNS zone, enter a command such as the following:

```
ServerIron(config)# show gslb dns zone foundrynet.com
```



The information is the same as the information displayed when you do not specify a zone name, except the ZONE field is unneeded and thus does not appear.

This display shows the following information.

**Table 9.5: Global SLB Zone and Host Application Information**

This Field...	Displays...
ZONE	<p>The zone name. The name that appears here is the name you specified when you configured the zone information.</p> <p><b>Note:</b> This field appears only if you do not specify the zone name when you display the information. If you specify the zone name, information for only that zone is displayed.</p>
HOST	<p>The host name. The name that appears here is the name you specified when you configured the host information.</p>
IP addresses	<p>The column of IP addresses lists the IP addresses the authoritative DNS server associated with the host name in the DNS reply. These are the servers that contain the content for the host. In this example, the servers contain the content for www.foundrynet.</p> <p>After evaluating the addresses using the GSLB policy, the GSLB ServerIron marks each address that passes the algorithm with an asterisk (*). An IP address that does not have an asterisk in front of it has not passed the GSLB algorithm and cannot be selected as the “best” site.</p> <p><b>Note:</b> If DNS override is enabled, only the addresses configured in the host’s IP list have asterisks and are valid choices for the best site. See “Enabling DNS Override” on page 9-31.</p>
Source	<p>The value following each server IP address indicates how the ServerIron learned the address. This field can have one of the following values:</p> <ul style="list-style-type: none"> <li>• c/g – The address is one that you associated with the host as part of the DNS override feature. See “Enabling DNS Override” on page 9-31.</li> <li>• d/c – The address was learned from the DNS server and also is one that you associated with the host.</li> <li>• dns – The address was learned from the DNS server.</li> </ul> <p>In the example above, the ServerIron learned about all the IP addresses associated with the zone name from the DNS server; thus, the source is listed as “dns”.</p>
Type	<p>The next value indicates the type of address, which can be one of the following:</p> <ul style="list-style-type: none"> <li>• v-ip – The address is a VIP configured on a ServerIron.</li> <li>• real-ip – The address is a real server.</li> </ul>

**Table 9.5: Global SLB Zone and Host Application Information (Continued)**

This Field...	Displays...
State	<p>The state of the server. The ServerIron determines the state based on the results of the Layer 7 health check(s) sent to the server. The ServerIron sends Layer 7 health checks for each host application you associate with the zone.</p> <p>The state can be one of the following:</p> <ul style="list-style-type: none"> <li>• <b>ACTIVE</b> – The server passed the Layer 4 and Layer 7 health checks and is presumed to be available.</li> <li>• <b>DOWN</b> – The server failed a health check. If any of the health checks are failed, the GSLB ServerIron disqualifies this site from being the “best” site.</li> </ul> <p><b>Note:</b> If the server has multiple applications, all the applications must pass the health check.</p> <p><b>Note:</b> The ServerIron also uses the results of the health check, if the server passes the check, in the TCP and App columns under FlashBack Delay, described below.</p>
Location	<p>The geographic location of the server. The location is based on the IP address and can be one of the following:</p> <ul style="list-style-type: none"> <li>• ASIA</li> <li>• EUROPE</li> <li>• N-AM – North America</li> <li>• S-AM – South America</li> </ul> <p>The GSLB ServerIron can use this information when comparing the servers in order to select the “best” ones for the client. The GSLB ServerIron prefers servers within the client’s geographic region over servers in other geographic regions.</p>
FlashBack Delay (x100us)	<p>The round-trip time for a health check sent by the GSLB ServerIron to the host application on the server.</p> <p>The GSLB ServerIron can use this information when comparing the servers in order to select the “best” ones for the client. The GSLB ServerIron prefers servers with lower round-trip times to those with higher round-trip times.</p> <p>The value in the TCP column indicates the round-trip time of the Layer 4 health check to the TCP port.</p> <p>The value in the App column indicates the round-trip time for the Layer 7 health check.</p> <p><b>Note:</b> A single value is displayed even if the zone has multiple host applications. If the FlashBack values (round-trip times) differ, the slowest times are displayed.</p>
DNS resp. selection percentage	<p>The percentage of times the GSLB ServerIron has selected this server as the “best” server and thus placed the server’s IP address at the top of the list in DNS replies.</p> <p><b>Note:</b> If you are using a ServerIron 400 and ServerIron 800, this field is called “DNS resp. selection counters”. See the next row.</p>

**Table 9.5: Global SLB Zone and Host Application Information (Continued)**

This Field...	Displays...
DNS resp. selection counters	<p>The number of times the GSLB ServerIron has selected this server as the “best” server. The percentage of times the GSLB ServerIron has selected this server as the best server also is shown.</p> <p><b>Note:</b> If you are using a ServerIronXL or ServerIronXL/G, this field is called “DNS resp. selection percentage”. See the previous row.</p>

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the DNS link to display the GSLB DNS information. See the previous section for descriptions of the information fields.

### Displaying Detailed DNS Zone Information

You can display all the information displayed by the **show gslb dns zone** command, plus information about the site and the ServerIron on which a VIP is configured, by entering the following command at any CLI level:

```
show gslb dns detail
```

This command is especially useful for sites that are configured for Symmetric Server Load Balancing. For information about this load balancing feature, see “Configuring Symmetric SLB and SwitchBack” on page 7-1.

This example assumes that the ServerIrons at the sunnyvale site are each configured with two VIPs for the “www” host and two VIPs for the “ftp” host in the foundrynet.com domain:

- VIPs 209.157.22.100 and 209.157.22.101 are configured on both ServerIrons for the “www” host.
- VIPs 209.157.22.102 and 209.157.22.103 are configured on both ServerIrons for the “ftp” host.

The same VIPs are configured on both ServerIrons, but only one of the ServerIrons is actively load balancing for a particular VIP. The other ServerIron is the standby for that VIP and assumes load balancing duties for the VIP only if the other ServerIron becomes unavailable. The default active ServerIron for a particular VIP is determined by the priority you assign to the VIP when you are configuring Symmetric SLB.

In this example, ServerIron slb-1 is the active ServerIron for VIPs 209.157.22.100 and 109.157.22.101 and ServerIron slb-2 is the default active ServerIron for VIPs 209.157.22.103 and 209.157.22.104. Although this example has both VIPs for a host active on the same ServerIron, you can just as easily configure the VIPs so that both ServerIrons have active VIPs for the same host.

**NOTE:** This example does not show the information for the atlanta site.

```

ServerIron(config)# show gslb dns detail
ZONE: foundrynet.com
HOST: www:

Flashback DNS resp.
delay      selection
(x100us)   percentage
TCP APP   (%)

* 209.157.22.227: dns      v-ip      ACTIVE N-AM.      6      60      40
                  site: sunnyvale, SI: slb-1 (209.157.22.209)
                  session util: 0%, avail. sessions: 524287
                  preference: 128

* 209.157.22.228: dns      v-ip      ACTIVE N-AM.      3      30      60
                  site: atlanta, SI: slb-1 (192.108.22.111)
                  session util: 10%, avail. sessions: 414269
                  preference: 128

* 210.224.100.5:  dns      real-ip   DOWN      ASIA      --      --      0
* 201.100.100.6:  dns      real-ip   DOWN      S-AM.     --      --      0
* 213.34.100.4:   dns      real-ip   DOWN      EUROPE    --      --      0

HOST: ftp:

Flashback DNS resp.
delay      selection
(x100us)   percentage
TCP APP   (%)

* 209.157.22.103: dns      v-ip      ACTIVE N-AM.      6      60      40
                  site: sunnyvale, SI: slb-2 (209.157.22.210)
                  session util: 7%, avail. sessions: 414287
                  preference: 128

* 209.157.22.104: dns      v-ip      ACTIVE N-AM.      3      30      60
                  site: atlanta, SI: slb-2 (192.108.22.112)
                  session util: 14%, avail. sessions: 324269
                  preference: 128

* 210.224.100.7:  dns      real-ip   DOWN      ASIA      --      --      0
* 201.100.100.8:  dns      real-ip   DOWN      S-AM.     --      --      0
* 213.34.100.9:   dns      real-ip   DOWN      EUROPE    --      --      0

```

**Syntax:** show gslb dns detail [<name>]

The <name> parameter specifies a zone name.

The text shown in bold type in the example is the information that is not displayed by the **show gslb dns zone** command.

**Table 9.6: Global SLB Zone and Host Application Information**

This Field...	Displays...
site	Indicates the site name of the ServerIron.
SI	Indicates the site ServerIron name and management IP address.
session util	Indicates the percentage of the ServerIron session capacity that is in use. This information is reported by the site ServerIron using the GSLB protocol.

**Table 9.6: Global SLB Zone and Host Application Information (Continued)**

This Field...	Displays...
preference	The numeric preference value for this site ServerIron. The preference can be used by the GSLB policy to select a site. See "Site ServerIron's Administrative Preference" on page 9-8.
avail. sessions	Indicates the number of unused sessions in the ServerIron's session table.

For descriptions of the other information displayed by the **show gslb dns detail** command, see "Displaying Zone and Host Name Information" on page 9-62.

#### *USING THE WEB MANAGEMENT INTERFACE*

You cannot display detailed zone and host information using the Web management interface.

## Displaying the GSLB Policy

You can display the default and user-configured settings for the GSLB policy. To do so, use either of the following methods.

### Displaying the Default GSLB Policy

To display the default GSLB policy, use either of the following methods.

#### *USING THE CLI*

To display the default GSLB policy, enter the following command:

```
ServerIron(config)# show gslb default
Default metric order: ENABLE
Metric processing order:
    1-Server health check
    2-Remote SI's session capacity threshold
    3-Round trip time between remote SI and client
    4-Geographic location
    5-Remote SI's available session capacity
    6-Server flashback speed
    7-Least response selection

DNS active-only: DISABLE   DNS best-only: DISABLE   DNS override: DISABLE
Modify DNS response TTL: ENABLE
DNS TTL: 10 (sec), DNS check interval: 30 (sec)
Remote SI status update period: 30 (sec)
Session capacity threshold: 90%, session capacity tolerance: 10%
Round trip time tolerance: 10%, round trip time explore percentage: 5%
Round trip time cache prefix: 20, round trip time cache interval: 120 (sec)
Flashback appl-level delay tolerance: 10%, TCP-level delay tolerance: 10%
```

**Syntax:** show gslb default

This display shows the following information.

**Table 9.7: GSLB Policy Information**

This Field...	Displays...
Default algorithm	<p>Indicates whether this policy is in effect. The value can be one of the following:</p> <ul style="list-style-type: none"> <li>• Disable</li> <li>• Enable</li> </ul> <p>If the state is Disable, then a user-configured policy is in effect instead.</p>
Metric processing order	<p>Indicates the order in which the selection metrics are applied to the server addresses in the DNS reply. For information about the metrics, see “The GSLB Policy” on page 9-5.</p>
DNS active-only	<p>Indicates whether the GSLB ServerIron removes IP addresses from the DNS response if those addresses fail a health check. This field can have one of the following values:</p> <ul style="list-style-type: none"> <li>• DISABLE – The ServerIron does not remove the IP addresses from the DNS response.</li> <li>• ENABLE – The ServerIron removes IP addresses that fail a health check from the DNS response.</li> </ul>
DNS best-only	<p>Indicates whether you have configured the ServerIron to remove all IP addresses except the “best” address from DNS replies. This field can have one of the following values:</p> <ul style="list-style-type: none"> <li>• DISABLE – The ServerIron does not remove all addresses except the best one.</li> <li>• ENABLE – The ServerIron removes all addresses except the best one.</li> </ul> <p><b>Note:</b> Even when this feature is enabled, if the GSLB policy does not result in selection of a best address, the DNS reply can still contain more than one address.</p> <p>For more information, see “Deleting All IP Addresses Except the Best One” on page 9-30.</p>
DNS override	<p>Indicates whether DNS override is enabled. DNS override replaces the addresses in a DNS reply with the “best” address from a list of addresses you configure. This field can have one of the following values:</p> <ul style="list-style-type: none"> <li>• DISABLE – The ServerIron does not replace the addresses in DNS replies with an address from a list you configure.</li> <li>• ENABLE – The ServerIron replaces the addresses in DNS replies with an address from a list you configure.</li> </ul> <p>For more information about DNS override, see “Enabling DNS Override” on page 9-31.</p>

Table 9.7: GSLB Policy Information (Continued)

This Field...	Displays...
Modify DNS response TTL	Indicates whether the GSLB ServerIron modifies the TTL in the DNS records in DNS responses before sending the responses to the client's DNS server. This field can have one of the following values: <ul style="list-style-type: none"> <li>DISABLE – The ServerIron does not modify the TTLs.</li> <li>ENABLE – The ServerIron modifies the TTLs.</li> </ul>
DNS TTL	Indicates the value (number of seconds) to which the GSLB ServerIron changes the TTL in each DNS record in the DNS responses before sending them to the client's DNS server.  <b>Note:</b> If the Modify DNS response TTL field contains "DISABLE", the ServerIron does not change the TTLs, regardless of the value in this field.
DNS check interval	Indicates how frequently the GSLB ServerIron refreshes its zone and host information with DNS servers.
Remote SI status update period	Indicates how frequently the remote ServerIrons send status updates to the GSLB ServerIron through the GSLB protocol.
Session capacity threshold	Specifies how close to its maximum session capacity the site ServerIron (remote ServerIron) can be and still be eligible as the best site for the client. If a site ServerIron exceeds the threshold, the site ServerIron is ineligible to be the best site.
Session capacity tolerance	Specifies the percentage by which the number of available sessions on the site ServerIron can differ from the number of available sessions on another site ServerIron and still be considered an equally good site. See "Site ServerIron's Available Session Capacity Tolerance" on page 9-7.
Round trip time tolerance	Specifies the percentage by which the RTT for one site can differ from the RTT for another site without this metric resulting in selection of one site over the other.
Round trip time explore percentage	Indicates the percentage of client requests from a given network for which the GSLB ServerIron intentionally ignores the RTT metric when evaluating the IP addresses in the DNS reply. The explore percentage prevents the ServerIron from continually biasing its site selection based on the first ServerIron to return RTT information. See "Modifying Round-Trip Time Values" on page 9-41.
Round trip time cache prefix	Indicates the length (number of significant bits) of entries in the GSLB ServerIron's IP address cache. The prefix determines the extent to which IP addresses are aggregated into entries in the cache.
Round trip time cache interval	Indicates how many seconds the GSLB ServerIron keeps an unrefreshed RTT cache entry in its cache before the entry ages out.
Flashback appl-level delay tolerance	Indicates the percentage of difference that can exist between application level FlashBack response times for two sites, without the ServerIron preferring one site over the other based on this metric.
TCP-level delay tolerance	Indicates the percentage of difference that can exist between Layer 4 FlashBack response times for two sites, without the ServerIron preferring one site over the other based on this metric.

**Table 9.7: GSLB Policy Information (Continued)**

This Field...	Displays...
Connection load limit	<p>Indicates the average load limit you specified using the <b>connection-load limit</b> &lt;average-load&gt; command. Each Interval:wt: value indicates the sampling interval and the weight assigned to the interval.</p> <p><b>Note:</b> This field applies only to ServerIronXLs running software release 07.3.04 or later.</p>

#### *USING THE WEB MANAGEMENT INTERFACE*

You cannot display the default GSLB policy parameters using the Web management interface.

#### **Displaying the User-Configured GSLB Policy**

To display the user-configured GSLB policy, use either of the following methods.

#### *USING THE CLI*

To display the user-configured GSLB policy, enter the following command:

```
ServerIron(config)# show gslb policy
Default metric order: ENABLE
Metric processing order:
    1-Server health check
    2-Remote SI's session capacity threshold
    3-Round trip time between remote SI and client
    4-Geographic location
    5-Remote SI's available session capacity
    6-Server flashback speed
    7-Least response selection

DNS active-only: DISABLE   DNS best-only: DISABLE   DNS override: DISABLE
Modify DNS response TTL: ENABLE
DNS TTL: 10 (sec), DNS check interval: 30 (sec)
Remote SI status update period: 30 (sec)
Session capacity threshold: 90%, session capacity tolerance: 10%
Round trip time tolerance: 10%, round trip time explore percentage: 5%
Round trip time cache prefix: 20, round trip time cache interval: 120 (sec)
Flashback appl-level delay tolerance: 10%, TCP-level delay tolerance: 10%
```

**Syntax:** show gslb policy



In this example, the default order of the policy metrics is in effect. Metrics that are disabled by default (such as the administrative preference) are not listed. In the following example, the order has been changed, two of the metrics have been disabled, and the administrative preference has been enabled.

```
ServerIron(config)# show gslb policy
Default metric order: DISABLE
Metric processing order:
    1-Round trip time between remote SI and client
    2-Remote SI's session capacity threshold
    3-Remote SI's available session capacity
    4-Server flashback speed
    5-Remote SI's preference value
    6-Least response selection

DNS active-only: DISABLE   DNS best-only: DISABLE   DNS override: DISABLE
Modify DNS response TTL: ENABLE
DNS TTL: 10 (sec), DNS check interval: 30 (sec)
Remote SI status update period: 30 (sec)
Session capacity threshold: 90%, session capacity tolerance: 10%
Round trip time tolerance: 10%, round trip time explore percentage: 5%
Round trip time cache prefix: 20, round trip time cache interval: 120 (sec)
Flashback appl-level delay tolerance: 10%, TCP-level delay tolerance: 10%
```

For a description of the information shown by this command, see “Displaying the Default GSLB Policy” on page 9-67.

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the Policy link to display the GSLB Policy panel, as shown in Figure 9.9 on page 9-27. See the previous section for descriptions of the information fields.

## Displaying RTT Prefix Cache Entries

The GSLB ServerIron maintains a cache of RTT information received from the site ServerIrons through the GSLB protocol. You can display the RTT information the GSLB ServerIron has related to a client IP address. To display the RTT information, specify a potential client address, as shown in the following example.

```
ServerIron(config)# show gslb cache 209.157.0.0
prefix length = 20, prefix = 209.157.0.0, region = N-AM
prefix source =          client-query

foundrynet.com:
  site = sunnyvale, SI = slb-1(209.157.22.209), rtt = 5 (x100 usec)
  site = atlanta, SI = slb-1(192.108.22.112), rtt = 10 (x100 usec)
```

The command in this example shows the RTT prefix information the GSLB ServerIron has related to client IP address 209.156.100.100. In this case, the GSLB ServerIron has two RTT entries for zone www.foundrynet.com.

**Syntax:** show gslb cache <ip-addr>

The <ip-addr> command specifies a site address.

The following example shows information for a user-configured static entry.

```
ServerIron(config)# show gslb cache 192.168.2.1
prefix length = 24, prefix = 192.168.2.0, region = N-AM
prefix source = static, client-query
www.foundrynet.com:
    site = atlanta, SI = slb-1(192.108.22.111), rtt = 5 (x100 usec)
```

This example shows the RTT prefix cache entry that contains site IP address 192.1678.2.1. The prefix source line indicates that the prefix cache entry that matches the site address was added statically. Notice that a prefix cache entry can have more than one source. In this case, the prefix was statically configured but a specific entry (listed below under the domain name “www.foundrynet.com”) was created when the GSLB ServerIron received RTT information from the site ServerIron for a site address within the prefix.

In the following example, a statically generated entry that the GSLB ServerIron created is displayed. The statically generated entries have an 8-bit prefix, whereas the prefix for dynamic entries is 20 bits long by default.

```
ServerIron(config)# show gslb cache 61.1.1.1

prefix length = 8, prefix = 60.0.0.0, region = ASIA
prefix source = geographic
```

This display shows the following information.

**Table 9.8: GSLB Policy Information**

This Field...	Displays...
prefix length	Specifies the length of the address prefix. The GSLB ServerIron initially generates the prefix tree using the IANA (geographic) allocated address prefixes, which have variable lengths. Dynamically generated cache entries (generated by client queries) have a fixed prefix length, as defined by the RTT cache-length parameter. The default is 20.
prefix	Specifies the prefix. All client addresses beginning with this prefix are aggregated in a single RTT entry.
region	Specifies the geographic location of this client prefix. This field can have one of the following values: <ul style="list-style-type: none"> <li>• ASIA</li> <li>• EUROPE</li> <li>• N-AM – North America</li> <li>• S-AM – South America</li> </ul>

Table 9.8: GSLB Policy Information (Continued)

This Field...	Displays...
prefix source	<p>Specifies the prefix source. This field can have one of the following values:</p> <ul style="list-style-type: none"> <li>geographic – displayed for static entries (entries generated by the GSLB ServerIron itself when initializing the RTT cache)</li> <li>client query – displayed for entries generated by client queries</li> <li>static – displayed for static entries entered by you (see “Configuring Static RTT Prefix Cache Entries” on page 9-43)</li> </ul> <p><b>Note:</b> If a static entry is long enough (greater than 20 bits) and has been accessed by a client query, the entry can show both sources.</p> <p>Notice that a prefix cache entry can have more than one source.</p>
site	Specifies the name of the site.
SI	Specifies the name and IP address of the ServerIron.
rtt	Specifies the RTT value.

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write or read-only access.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to GSLB in the tree view to expand the list of Global SLB option links.
4. Select the Cache link.
5. Enter an IP address in the Enter IP Address field, then click Apply. If the GSLB ServerIron has cache entries for the specified address, the entries are listed. See the previous section for a description of the information fields.

### Displaying GSLB Resource Utilization

For GSLB parameters, you can display the number of currently configured items and the maximum number of items you can configure on the ServerIron. To display this information, use the following CLI method.

#### USING THE CLI

To display GSLB resource information, enter the following command at any level of the CLI:

```
ServerIron(config)# show gslb resources
GSLB resource usage:
```

	Current	Maximum
sites	1	100
SIs	2	200
SIs' VIPs	2	2000
dns zones	2	200
dns hosts	2	400
health-checks app.	2	600
dns IP addrs.	5	2000
affinities	0	50
static prefixes	4	250
prefix cache	104	5050
RTT entries	1	10000

The values in the Current column indicate how many of each GSLB configuration or data item are currently on the GSLB ServerIron. The values in the Maximum column list the maximum number of each item the GSLB ServerIron can hold.

This command shows the following information.

**Table 9.9: GSLB Resources**

This Field...	Displays...
sites	The number of remote sites configured on the GSLB ServerIron.
SIs	The number of remote site ServerIrons configured on the GSLB ServerIron. Each remote site ServerIron is associated with a site. When you add a remote site ServerIron, the GSLB ServerIron uses the GSLB protocol to establish a TCP session with port 182 on the remote ServerIron, for gathering information to use with the GSLB policy.
SIs' VIPs	The number of virtual IP addresses (VIPs) configured on the site ServerIrons that the GSLB ServerIron has cached, and the maximum number of site VIPs the GSLB ServerIron can cache.
dns zones	The number of zone names currently configured on this GSLB ServerIron and the maximum number that can be configured.
dns hosts	The number of host names currently configured on this GSLB ServerIron and the maximum number that can be configured.
health-checks app.	The number of applications currently configured on this GSLB ServerIron and the maximum number that can be configured. The ServerIron performs Layer 4 and, if applicable, Layer 7 health checks on each application.
dns IP addrs.	The number of IP addresses the GSLB ServerIron has learned from the DNS server, and the maximum number of DNS records the GSLB ServerIron can store in memory.
affinities	The number of affinity definitions currently configured on the GSLB ServerIron and the maximum number that can be configured.
static prefixes	The number of statically configured prefixes in the GSLB ServerIron's prefix cache, and the maximum number of statically configured prefixes the cache can hold. For information, see "Configuring Static RTT Prefix Cache Entries" on page 9-43.
prefix cache	The total number of prefixes currently in the prefix cache, and the maximum number the cache can hold. The prefix entries include static ones used for geographic information, user-configured prefixes, and dynamic prefixes created when client queries are received. Dynamic entries age out when unused.

**Table 9.9: GSLB Resources (Continued)**

This Field...	Displays...
RTT entries	The number of cached per-prefix, per-domain name RTT records. For each client prefix, the GSLB ServerIron stores the most recently accessed domain names (up to 10 per client, ordered from most to least recent). For each domain name the GSLB ServerIron stores the site the GSLB ServerIrons that currently have the best RTT to the client prefix (up to four such the GSLB ServerIrons: two current best choices plus two potentials). The GSLB ServerIron has separate records for each domain name because the closest site can be different for different domain names (unless every remote ServerIron serves every domain name). If the maximum is reached, the GSLB ServerIron stops creating new records.

*USING THE WEB MANAGEMENT INTERFACE*

You cannot display the GSLB resource information using the Web management interface.

**Displaying Dynamic Configuration Information**

When you configure GSLB, the ServerIron creates dynamic real server configurations based on the IP addresses the GSLB ServerIron receives in response to DNS queries sent to the real DNS server. These real servers are created for health check purposes only, and do not play a role in SLB. In the dynamic configurations, the site IP addresses contained in DNS replies are the names and IP addresses of the real servers. The ServerIron creates internal virtual servers and then binds the dynamic real servers to the virtual servers based on the application ports you specify when you add GSLB zones and hosts.

This information can be useful when troubleshooting your GSLB configuration, by showing you the internal servers and port bindings the ServerIron created based on your GSLB configuration. For example, if your configuration uses multiple zone names associated with the same IP address, you can verify that the ServerIron created an alias TCP port number for each additional zone and application associated with the IP address.

To display the dynamic server configuration information, use the following CLI methods.

*USING THE CLI*

The commands and displays for dynamic server configuration information are based on the commands and displays for SLB server configuration information. You can display the following dynamic configuration information:

- Real server information – There is one dynamically created real server per IP address.
- Virtual server information – There is one dynamically created virtual server per domain name.
- Port binding information – The TCP and UDP ports
- Session statistics – Another way to list the real servers.

To display dynamic server information, enter the commands shown in the following examples. The portions of the output that are shown in bold type are those of interest.

The **show server dynamic real** command shows the real servers that the ServerIron dynamically has dynamically created for the site addresses from DNS replies.

```
ServerIron(config)# show server dynamic real
Real Servers Info

Name : 209.157.22.229
IP:209.157.22.229 Range:1 State:Active Wt:1 Max-conn:1000000

Port      State      Ms CurConn TotConn Rx-pkts Tx-pkts Rx-octet Tx-octet Reas
----      -
http      active      0 0      0      0      0      0      0      0
default  unbnd      0 0      0      0      0      0      0      0
Server  Total      0      0      0      0      0      0      0

Name : 209.157.22.230
IP:209.157.22.130 Range:1 State:Active Wt:1 Max-conn:1000000

Port      State      Ms CurConn TotConn Rx-pkts Tx-pkts Rx-octet Tx-octet Reas
----      -
http      failed      0 0      0      0      0      0      0      0
default  unbnd      0 0      0      0      0      0      0      0
Server  Total      0      0      0      0      0      0      0
```

This example shows real servers dynamically created for two sites that were listed in DNS replies.

The **show server dynamic virtual** command shows internal virtual servers. The purpose of these servers is to allow the ServerIron to bind to the dynamically created real servers. The ServerIron uses private IP addresses starting with 10.10.10.10 for the names and IP addresses of the virtual servers. The ServerIron does not respond to pings or ARP requests to the addresses it uses for the internal virtual servers. Thus, if your network also uses these addresses, the virtual servers do not create address conflicts.

---

**NOTE:** Since the dynamic virtual servers use addresses in the 10.10.10.x/23 subnet for the internal database, a GSLB ServerIron cannot support user-configured real and virtual servers in this address range.

---

```
ServerIron(config)# show server dynamic virtual
```

#### Virtual Servers Info

```

Server Name: 10.10.10.10      IP : 10.10.10.10      : 1
Status: enabled Predictor: round-robin TotConn: 0
Dynamic: Yes HTTP redirect: disabled
ACL: id = 0
Sym: group = 1 state = 5 priority = 0 keep = 0
Activates = 1, Inactive= 0

Port    State    Sticky  Concur  Proxy    CurConn  TotConn  PeakConn
http    enabled  NO      NO      NO        0         0         0
default enabled  NO      NO      NO        0         0         0

Server Name: 10.10.10.11      IP : 10.10.10.11      : 1
Status: enabled Predictor: round-robin TotConn: 0
Dynamic: Yes HTTP redirect: disabled
ACL: id = 0
Sym: group = 1 state = 5 priority = 0 keep = 0
Activates = 1, Inactive= 0

Port    State    Sticky  Concur  Proxy    CurConn  TotConn  PeakConn
70      enabled  NO      NO      NO        0         0         0
default enabled  NO      NO      NO        0         0         0

```

The **show server dynamic bind** command shows the port bindings the ServerIron creates for the dynamically created real servers and virtual servers. This example shows that the ServerIron has bound internal virtual server 10.10.10.10 to real server 209.157.22.229 for TCP port 80 (HTTP). The ServerIron also has bound internal virtual server 10.10.10.11 to real server 209.157.22.230, using TCP port 80.

```

ServerIron(config)# show server dynamic bind
Virtual Server Name: 10.10.10.10, IP: 10.10.10.10
    http -----> 209.157.22.229: 209.157.22.229, http (remote)
Virtual Server Name: 10.10.10.11, IP: 10.10.10.11
    70 -----> 209.157.22.230: 209.157.22.230, 70 (remote)

```

The **show server dynamic sessions** command provides a simple way to list the real servers. The output is based on the output for the **show server sessions** command. However, in the case of dynamically created servers, there are no meaningful session statistics in this display.

```

ServerIron(config)# show server dynamic sessions
Avail. Sessions      =      524287  Total Sessions      =      524288
Total C->S Conn      =          90  Total S->C Conn      =          0
Total Reassign       =          0  Unsuccessful Conn    =          2
Server State - 1:enabled, 2:failed, 3:test, 4:suspect, 5:grace_dn, 6:active

Real Server    State  CurrConn  TotConn  TotRevConn  CurrSess  PeakConn
209.157.22.229    6        0         0         0         0         0
209.157.22.230    6        0         0         0         0         0

```

## SNMP Traps and Syslog Messages

The ServerIron software can generate the following SNMP traps and Syslog messages related to GSLB. All traps and Syslog messages are enabled by default.

- GSLB ServerIron events:
  - Establishment of the GSLB protocol connection between the GSLB ServerIron and the remote ServerIron
  - Termination of the GSLB protocol connection between the GSLB ServerIron and the remote ServerIron

Remote site ServerIron events:

- Establishment of the GSLB protocol connection between the remote ServerIron and the GSLB ServerIron
- Termination of the GSLB protocol connection between the remote ServerIron and the GSLB ServerIron
- Health-check events:
  - The GSLB ServerIron determines that the IP address for a domain name is active
  - The GSLB ServerIron determines that the IP address for a domain name is down
  - A TCP or UDP port passes the Layer 4 health check and its status changes to “active”
  - A TCP or UDP port fails the Layer 4 health check and its status changes to “down”

---

**NOTE:** All the health check events are on the GSLB ServerIron, not on the remote site ServerIron.

---

A given domain name can be associated with multiple health check TCP or UDP ports. In that case, the GSLB ServerIron considers an IP address to be active only if all its associated TCP and UDP ports pass their health checks.

State transitions of individual ports are determined as a part of the health check procedure. However, state transitions for IP addresses are detected during GSLB decision making (when the GSLB ServerIron is processing a DNS response or when you display zone information). In these cases, health check status changes affect the GSLB decisions.



## Displaying the Syslog Messages

By default, the ServerIron's Syslog buffer is enabled and contains up to 50 entries. To display the GSLB and other Syslog messages, enter the following command at any level of the CLI:

```
ServerIron> show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Buffer logging: level ACDMEINW, 16 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Dynamic Log Buffer (50 entries):
00d00h01m28s:N:L4 gslb health-check 209.157.22.210 of foundrynet.com status
changed to up
00d00h01m28s:N:L4 gslb health-check 209.157.22.209 of foundrynet.com status
changed to up
00d00h01m01s:I:Interface ethernet16, state up
00d00h01m01s:I:Interface ethernet2, state up
00d00h00m34s:N:L4 gslb health-check 209.157.22.210 of foundrynet.com port 80 is up
00d00h00m32s:N:L4 gslb health-check 209.157.22.209 of foundrynet.com port 80 is up
00d00h00m32s:N:L4 server 209.157.23.59 kandrew TCP port 80 is up
00d00h00m32s:N:L4 server 209.157.23.59 kandrew is up
00d00h00m31s:N:L4 gslb connection to site sunnyvale SI 209.157.22.210 slb-2 is up
00d00h00m31s:N:L4 gslb connection to site sunnyvale SI 209.157.22.209 slb-1 is up
00d00h00m31s:I:Bridge topology change, vlan 1, interface 16, changed state to
forwarding
00d00h00m07s:N:L4 server 209.157.23.130 dns-ivy TCP port 53 is up
00d00h00m07s:N:L4 server 209.157.23.130 dns-ivy is up
00d00h00m06s:I:Bridge topology change, vlan 1, interface 2, changed state to
forwarding
00d00h00m03s:I:Bridge root changed, vlan 1, new root ID 800000e0520002d1, root
interface 16
00d00h00m00s:I:Warm start
```

In this example, the GSLB messages are shown in bold type. The GSLB messages in this example all apply to the ServerIrons at the Sunnyvale site. Three types of messages are shown.

- The first two GSLB messages are shown nearest the bottom, since new messages appear at the top. These two messages indicate that the GSLB ServerIron has established a GSLB protocol connection to the site ServerIrons (slb-1 at 209.157.22.209 and slb-2 at 209.157.22.210).
- The next two GSLB messages indicate that the Layer 4 health checks for TCP port 80 were completed successfully. For sites with other applications, the ServerIron sends separate Layer 4 TCP or UDP health checks for each of those applications.
- The final two GSLB messages in this example (the ones nearest the top of the log) indicate that the site ServerIrons responded to the Layer 3 health check (IP ping).

## Disabling and Re-Enabling Traps

All traps, including GSLB traps, are enabled by default. To disable a GSLB trap, use either of the following methods.

### USING THE CLI

To disable a GSLB trap, enter a command such as the following:

```
ServerIron(config)# no snmp-server enable traps l4-gslb-remote-si-down
```

The command in this example disables the trap that occurs if a remote site ServerIron fails its Layer 3 health check and its status therefore changes from "up" to "down".

**Syntax:** [no] snmp-server enable traps <trap-type>

For GSLB, the trap type can be one of the following:

- **I4-gslb-remote-gslb-si-down** – Generated when the GSLB protocol connection from this site ServerIron to a remote GSLB ServerIron goes down.
- **I4-gslb-remote-gslb-si-up** – Generated when the GSLB protocol connection from this site ServerIron to a remote GSLB ServerIron comes up.
- **I4-gslb-remote-si-down** – Generated when the GSLB protocol connection from this GSLB ServerIron to a remote site ServerIron goes down.
- **I4-gslb-remote-si-up** – Generated when the GSLB protocol connection from this GSLB ServerIron to a remote site ServerIron comes up.
- **I4-gslb-health-check-ip-down** – Generated when GSLB determines that the IP address belonging to a domain name for which the ServerIron is providing GSLB is down.
- **I4-gslb-health-check-ip-up** – Generated when GSLB determines that the IP address belonging to a domain name for which the ServerIron is providing GSLB is now active.
- **I4-gslb-health-check-ip-port-down** – Generated when an application port in a domain on the site IP address fails its Layer 4 TCP or UDP health check, resulting in a status change to “down”.
- **I4-gslb-health-check-ip-port-up** – Generated when an application port in a domain on the site IP address passes its Layer 4 TCP or UDP health check, resulting in a status change to “up”.

#### **USING THE WEB MANAGEMENT INTERFACE**

1. Log on to the device using a valid user name and password for read-write access. The System configuration panel is displayed.
2. Select the Management link to display the Management panel.
3. Select the Disable or Enable button next to the trap you want to disable or enable.
4. Click the Apply button to save the change to the device's running-config file.
5. Select the Save link at the bottom of the dialog. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

---

# Chapter 10

## Configuring Transparent Cache Switching

This chapter describes how to configure Transparent Cache Switching (TCS) on the ServerIron and Foundry switches using the Command Line Interface (CLI) and Web management interface. See the *Foundry ServerIron Command Line Interface Reference* for information about the CLI commands.

To display TCS configuration information and statistics, see “Displaying Configuration Information and Statistics” on page 10-27.

Application examples at the end of the chapter show practical applications of the TCS features. See “TCS Application Examples” on page 10-32.

### Configuring TCS

By default, TCS is disabled. To use TCS, perform the following tasks:

1. Enable the web cache feature on either a global (switch) or local (interface) basis by configuring a cache policy.

---

**NOTE:** You cannot enable the web cache feature on both a global (switch) and local (interface) basis.

---

2. Assign IP addresses for each web cache server.
3. Assign web cache servers to specific cache groups.
4. Assign an interface to a cache group (optional).
5. Define distribution of web requests within a cache group (optional).
6. Modify default settings for TCS parameters (optional).
7. Define IP filters to manage the use of caching on the network (optional).
8. Save the configuration to flash.

## TCS Parameters

Table 10.1 on page 10-2 lists the TCS parameters and provides a page reference for more information about each feature.

**Table 10.1: ServerIron TCS Parameters**

Feature	See page...
<b>ServerIron Global Parameters (apply to both SLB and TCS)</b>	12-1 10-31 6-30
Ping interval and retries (3 Layer health check)	12-19
HTTP keepalive interval and retries	12-32
Force shutdown option	10-17
Additional IP addresses (for source Network Address Translation)	6-30
<b>TCS Global Parameters</b>	10-16
TCS state (enabled or disabled)	10-4
Cache Route Optimization (CRO)	10-16
<b>Cache Group Parameters</b>	10-7
Cache group	10-8
Hash-distribution values	10-12
Policy-based Cache Failover (CFO)	10-14
Cache server IP address spoofing	10-14
<b>Cache Server Parameters</b>	10-17
Cache server	10-7
Maximum connections	10-18
Weight	10-18
HTTP keepalive method (GET or HEAD), value, and status codes	C-1
FastCache (for asymmetric topologies)	10-19
Destination NAT	10-19
Source NAT	10-20
Remote Cache	10-22
Policy-based cache switching	10-22
<b>TCP/UDP Application Port Parameters</b>	10-23
Port priority	10-23
Health checks	12-1 12-3

**Table 10.1: ServerIron TCS Parameters (Continued)**

<b>Feature</b>	<b>See page...</b>
Connection Rate Limiting (CRL)	10-24
<b>Port (physical interface) Parameters</b>	10-11
Cache group membership; adding an interface to a cache group	10-11
Cache group membership; removing an interface from a cache group	10-11

### Configuration Notes

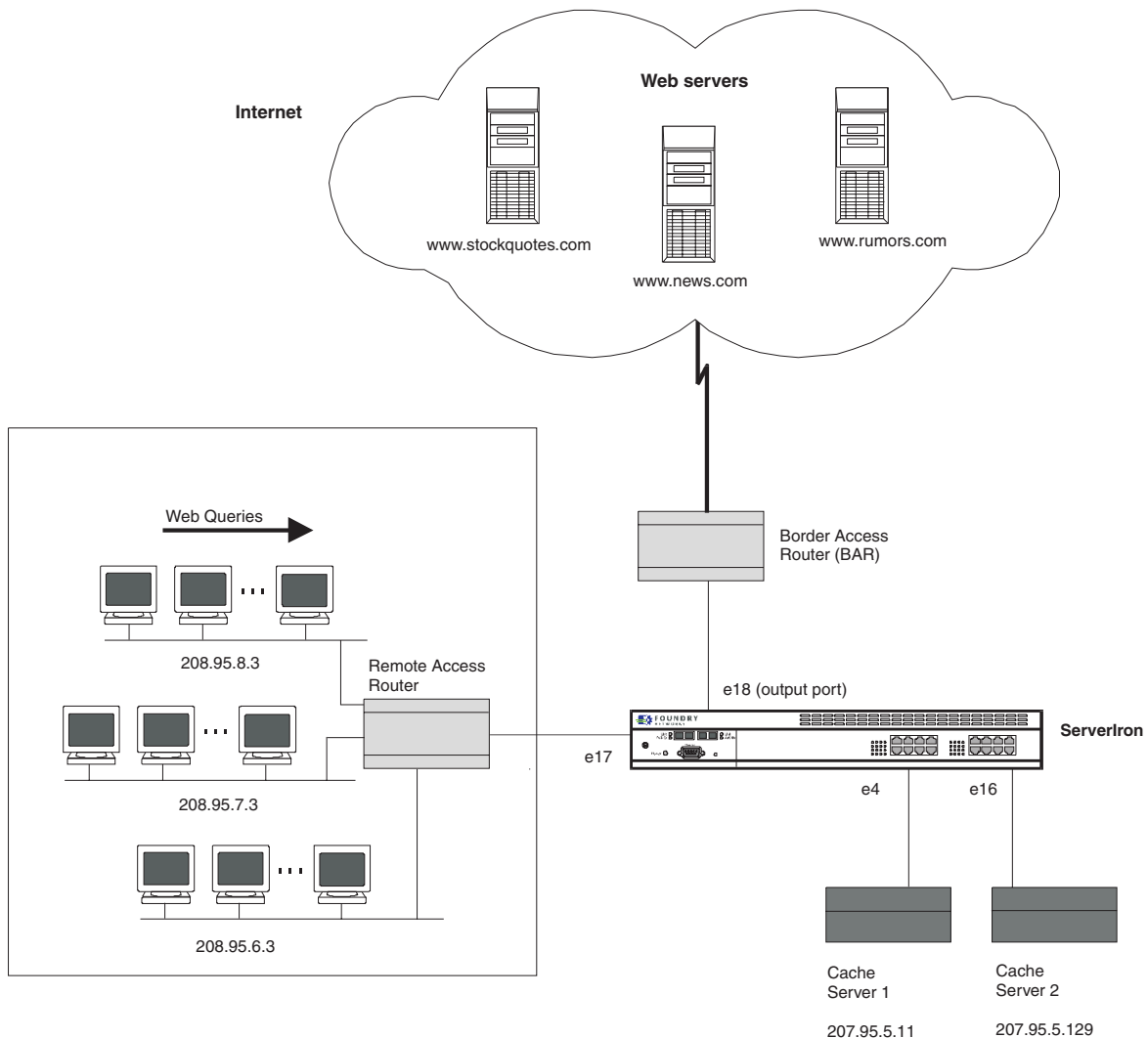
- Once TCS is enabled on a switch, *all* ports on the switch are members of cache group 1 by default.
- You can configure up to four cache groups. The default group is 1.
- Web cache servers must be members of a cache group.
- A cache group is defined in terms of input ports to the ServerIron. To give a client access to a group of cache servers, the input port connecting the client to the ServerIron must be in the cache group that contains the cache servers. If you plan to have only one cache group, you do not need to add the input ports to the cache group because all ports are members of cache group 1 by default.
- If you do not want a specific port to support TCS (for example, you want to redirect HTTP traffic for that port directly to the Internet instead), then you need to remove that port from default cache group 1. For more information, see “Removing an Interface from a Cache Group” on page 10-11.
- You must apply an IP policy to redirect Internet traffic to the cache servers. You can apply a global or local policy. A global policy takes effect on all ports as soon as you configure the policy. A local policy does not take effect until it is assigned to an interface. If you assign a local policy, assign it to the output port connected to the Internet. The policy sends all HTTP traffic addressed as output traffic to the port to the CPU instead for processing and forwarding. For example, in Figure 10.1, interface e18 has a locally defined policy.

### Example TCS Application

Suppose you want to use transparent caching within the network to increase the performance of web queries and lessen the demands on the current WAN access link to the Internet.

Two cache servers, server1 and server2, are installed within the network to handle the transparent caching of web (HTTP) traffic. TCS is enabled on the ServerIron to direct all HTTP traffic to the cache servers for processing, as shown in Figure 10.1.

**Figure 10.1 Using transparent caching within a ServerIron network**



## Enabling TCS

When TCS is enabled, the feature detects HTTP traffic addressed for output to the Internet and redirects the traffic to the CPU, which processes the traffic and forwards it to the cache servers instead.

TCS is assigned as a Layer 4 QoS priority type. HTTP traffic can be transparently defined on a switch on either a global (switch) or local (interface) basis. If HTTP traffic is transparently defined on a global basis, all ports redirect HTTP traffic toward the cache servers; however, only the port connected to the Internet router forwards traffic out toward the Internet (for example, port 18 in Figure 10.1).

---

**NOTE:** You cannot enable TCS on both a global (switch) and local (interface) basis.

---

The value of defining TCS on all ports (globally) is expediency. Globally assigning TCS to all ports eliminates the need to individually configure ports added in the future.

---

**NOTE:** HTTP ports are automatically created and enabled when a web cache server is created on the ServerIron, so there is no need to assign HTTP ports on `server1` and `server2` of this example.

---

### USING THE CLI

To enable TCS on all interfaces (globally) of the ServerIron shown in Figure 10.1, enter the following command:

```
ServerIron(config)# ip policy 1 cache tcp 80 global
```

**Syntax:** ip policy <index> cache | normal | high tcp | udp <tcp/udp-portnum> global | local

---

**NOTE:** The value '1' in the example above refers to the index value for the policy. This number can be any unused number from 1 – 64. Thus, up to 64 IP policies can be defined on a switch.

---

To enable transparent cache switching of HTTP traffic for e18 only, as opposed to globally on all of the ports, enter the following commands:

```
ServerIron(config)# ip policy 2 cache tcp 80 local
```

```
ServerIron(config)# int e 18
```

```
ServerIron(config-if-18)# ip-policy 2
```

**Syntax:** ip-policy <index>

---

**NOTE:** If you want a local policy to be supported, you must first configure it globally and then assign it on the interface level, as shown in example 2 above.

---

---

**NOTE:** Because only port e18 is connected to an Internet router in the configuration shown in Figure 10.1, you can accomplish the same result by enabling TCS on a local basis for port e18, as shown in example 2.

---

### USING THE WEB MANAGEMENT INTERFACE

To globally enable TCS on a ServerIron:

1. Log on to the device using a valid user name and password for read-write access.

---

**NOTE:** Beginning with software release 05.0.00, the software does not have a default read-write SNMP community. If you use the default community name "private" as the password for Web management access, you need to use the CLI to add the read-write community string first. See "Establishing SNMP Community Strings" on page 3-11.

---

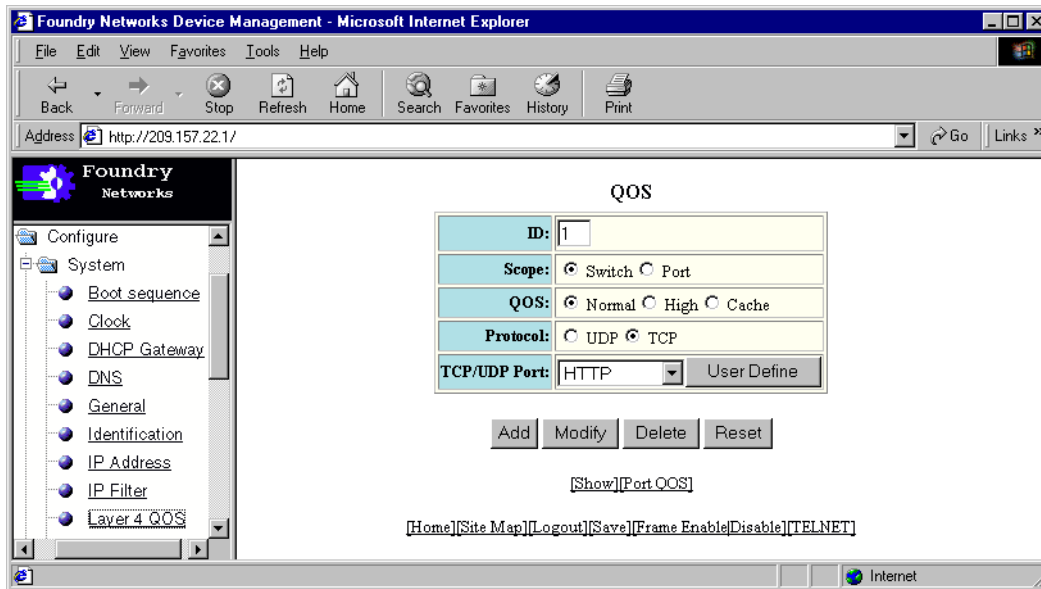
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to System in the tree view to expand the list of system configuration links.
4. Select the Layer 4 QOS link to display a panel such as the one shown in Figure 10.2.

---

**NOTE:** If any QoS priorities have been assigned on the switch, the QoS summary panel will appear first. Select the Add QOS link.

---

**Figure 10.2 Globally enabling TCS on a ServerIron**



5. Enter the ID number for the IP policy profile to be defined. Possible values are from 1 – 64.

---

**NOTE:** IP policies control QoS as well as filter on IP addresses and TCP/UDP ports.

---

6. Select a scope type of either switch or port.

---

**NOTE:** If you want to apply TCS to all ports on the system, select the switch option. If you want to apply TCS to an individual port only, select the port option.

---

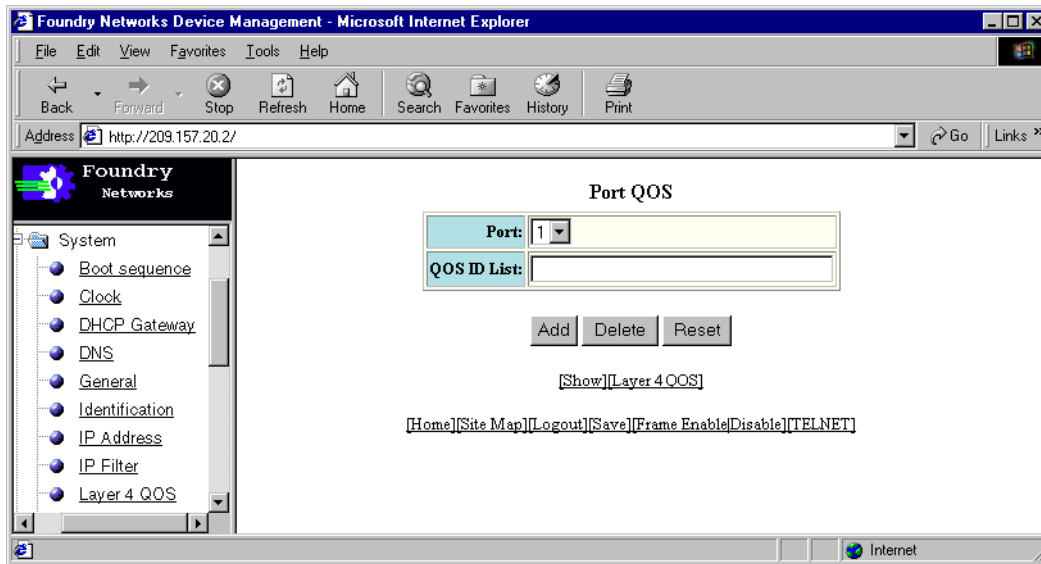
7. Select Cache as the QoS option.
8. Select TCP as the protocol.
9. Select HTTP from the TCP/UDP port pulldown menu.
10. Select the Add button to enable TCS for the switch or port.
11. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

To locally enable TCS on an interface, after enabling the feature globally:

1. Complete steps 1 – 11 above to enable the TCS feature globally.
2. Select the Port QoS link from the QoS Configuration panel. The panel shown in Figure 10.3 will appear.



Figure 10.3 Enabling TCS locally on an interface



3. Select the port that TCS is to be assigned to from the pulldown menu.
4. Enter the ID of the TCS priority assigned on the QoS panel on which you enabled TCS on the system globally. In this example, enter the value 1.
5. Select the Add button to assign TCS to the selected interface.
6. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Adding Web Cache Servers

Once you have enabled TCS on the ServerIron, you need to assign a name and IP address to each cache server. Once you have assigned the name and IP address, you can reference the server in CLI commands by either the server's name or its IP address.

The name for the server can be any alphanumeric string of up to 32 characters. The IP address entered is the IP address of the web cache server.

### USING THE CLI

To assign the names and IP addresses to the cache servers shown in Figure 10.1:

```
ServerIron(config)# server cache-name server1 207.95.5.11
ServerIron(config-rs-server1)# server cache-name server2 207.95.5.129
ServerIron(config-rs-server2)# end
ServerIron# write mem
```

**Syntax:** server cache-name <text> <ip-addr>

### USING THE WEB MANAGEMENT INTERFACE

To assign names and IP addresses to the cache servers:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to TCS in the tree view to expand the list of Transparent Cache Switching option links.

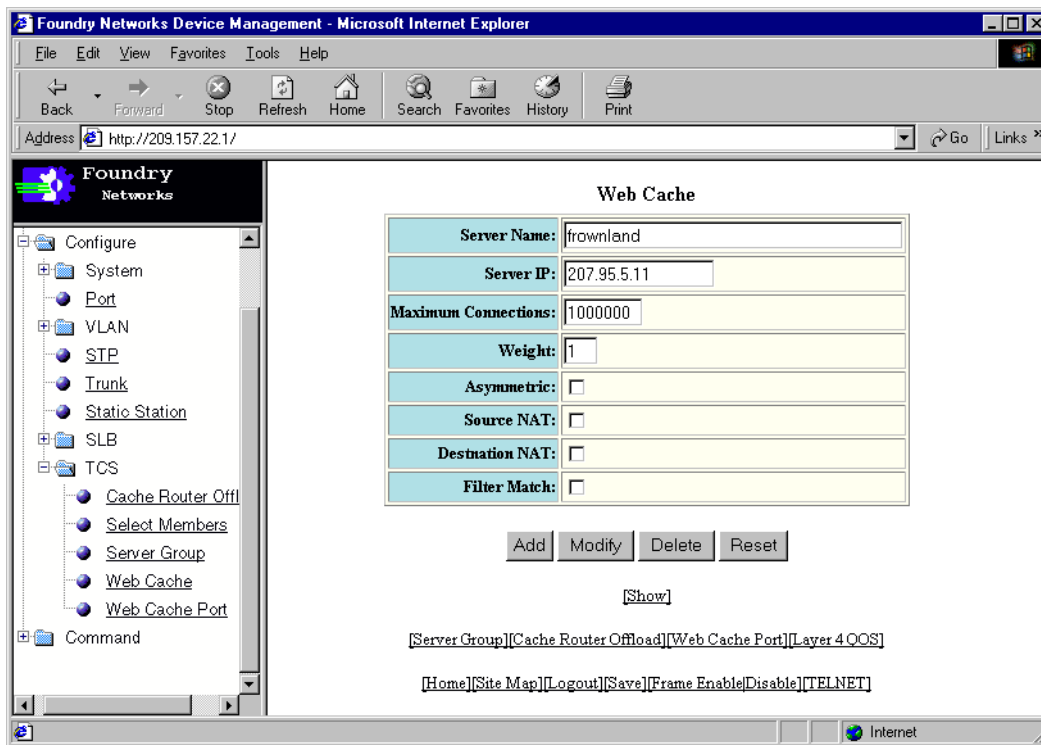
4. Select the Web Cache link from the menu. A panel such as the one in Figure 10.4 will appear.

---

**NOTE:** If cache servers are already configured on the ServerIron, a summary panel listing the configured cache servers is listed instead. In this case, select the Add Web Cache link.

---

**Figure 10.4** Web Cache Add panel



5. Enter the name of the server in the Server Name field.
6. Enter the IP address in the Server IP field.
7. Change other parameters from their default values if needed. (See “Modifying Default Settings for Cache Servers” on page 10-17 for information on changing these parameters.)
8. Select the Add button to assign the name and IP address for the cache server.
9. Repeat steps 5 – 8 for each cache server you want to define.
10. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device’s flash memory.

## Assigning Web Cache Servers to a Cache Group

All cache servers must belong to a cache group. The ServerIron uses a hashing algorithm to distribute HTTP requests among the servers in the cache group. (See “Defining Distribution of Web Requests Within a Cache Group” on page 10-12.) In addition, cache groups provide automatic recovery from a failed or otherwise out-of-service web cache server. If a web cache server failure occurs, ServerIron detects the failure and directs subsequent requests to the next available cache server or forwards the request directly to the WAN link. Up to four server cache groups can be assigned to a Foundry switch.

---

**NOTE:** All cache servers must be assigned to a cache group.

---

**NOTE:** You can gain additional reliability by using redundant ServerIrons, thus eliminating any single point of failure in the network path to the web cache server group. For more details on configuring redundant switches using the server backup option, see “Configuring Hot Standby Redundancy” on page 5-1.

### USING THE CLI

To assign cache servers 1 and 2 to the same cache group (as in Figure 10.1), you first create the server group and then assign the servers to the group, as shown in the following example:

```
ServerIron(config)# server cache-group 1
ServerIron(config-tc-1)# cache-name server1
ServerIron(config-tc-1)# cache-name server2
```

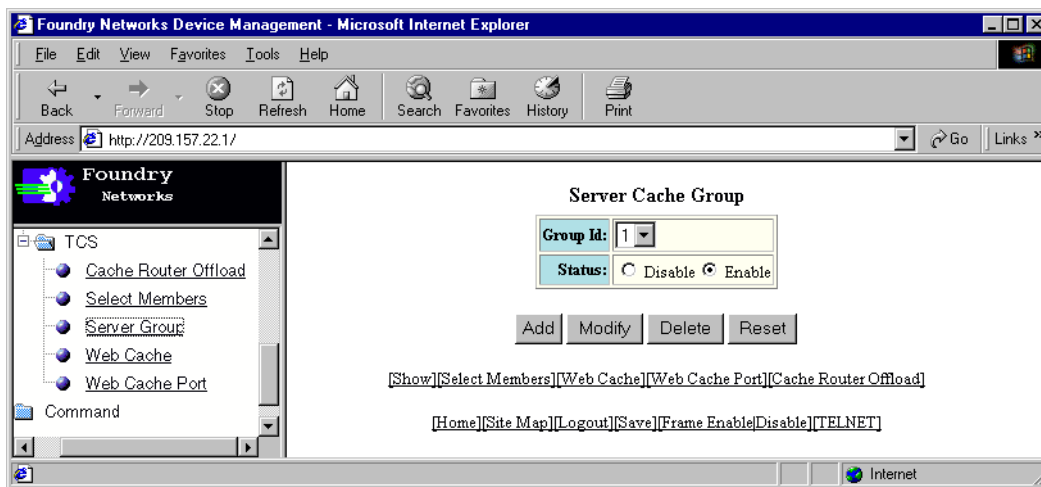
**NOTE:** The **server cache-group 1** CLI command launches you into the Cache Group Level of the CLI. Assign the cache servers to the server group at the Cache Group Level.

### USING THE WEB MANAGEMENT INTERFACE

To assign cache servers a cache group, create a server group and then assign servers to that group, as shown in the following example:

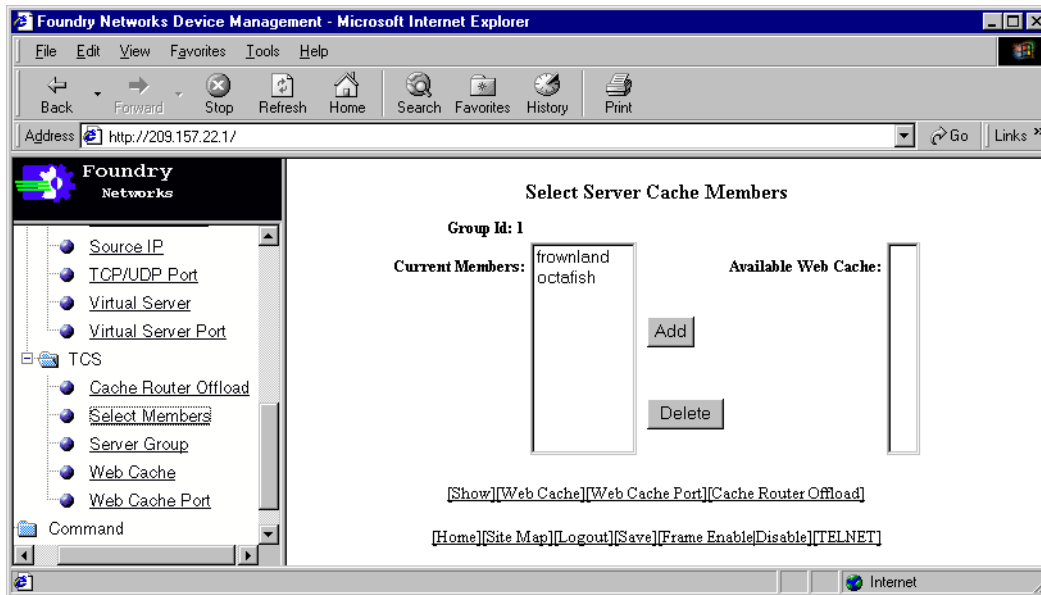
1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to TCS in the tree view to expand the list of Transparent Cache Switching option links.
4. Select the Server Group from the menu. The panel shown in Figure 10.5 will appear.

**Figure 10.5** Cache server group assignment panel



5. Select a group ID from the pulldown menu. Possible values are 1 – 4.
6. Select the Add button to assign the change.
7. To assign cache servers to that group, choose the Select Members link. The panel shown in Figure 10.6 will appear.

Figure 10.6 Select Server Cache Members panel



8. Highlight the servers listed in the Available Web Cache column that you want to assign to the cache group.
9. Select the Add button next to the Current Members column. The highlighted servers are added to the cache group.
10. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Disabling a Cache Group or a Server Within a Cache Group

You can disable a cache group or server to allow for maintenance.

### USING THE CLI

To disable cache group 1, enter the following commands:

```
ServerIron(config)# server cache-group 1
ServerIron(config-tc-1)# disable
```

**Syntax:** [no] disable

To disable a server (server2) within an active cache group, enter the following commands at the cache server level:

```
ServerIron(config)# server cache-name server2
ServerIron(config-rs-server2)# port http disable
```

**Syntax:** port <TCP/UDP port> disable

---

**NOTE:** For TCS, the only supported TCP/UDP port is HTTP.

---

### USING THE WEB MANAGEMENT INTERFACE

To disable a cache group:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to TCS in the tree view to expand the list of Transparent Cache Switching option links.

4. Select the Server Group link from the menu.
5. Select the Modify button for the server cache group to be disabled.
6. Select disable next to the Status option.
7. Select the Modify button to assign the change.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Removing or Re-Assigning an Interface

By default, all ports (physical interfaces) on the ServerIron belong to cache group 1. You can remove an interface from cache group 1 by assigning it to another cache group or by explicitly removing the port from cache group 1. An interface can be in only one cache group. If you assign an interface to a cache group, the ServerIron automatically removes that interface from cache group 1 or whatever cache group the interface is in.

### Removing an Interface from a Cache Group

You can remove an interface from a cache group to assign it to another cache group or to bias its traffic away from cache servers entirely.

Suppose you want to direct all web traffic sent from interface 3 of the ServerIron directly to the Internet and to bypass the cache servers within the network. Because all interfaces are by default assumed to be a member of cache group 1, you must remove interface 5 from cache group 1.

#### USING THE CLI

```
ServerIron(config)# interface ethernet 3
ServerIron(config-if-3)# no cache-group 1
```

**Syntax:** [no] cache-group 1

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to System in the tree view to expand the list of system configuration links.
4. Select the Port link from the menu. The summary Port panel will appear.
5. Select the Modify button next to the entry for interface 3.
6. Select "none" from the Cache Group ID pulldown menu.
7. Select the Apply button to remove interface 3 from cache group 1.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Assigning an Interface to a Cache Group

You can define how HTTP traffic from a specific interface is handled by assigning it to a specific cache group. By default, each interface is already a member of cache group 1.

#### USING THE CLI

To assign interface 3 to cache group 1, enter the following commands:

```
ServerIron(config)# interface ethernet 3
ServerIron(config-if-3)# cache-group 1
```

---

**NOTE:** You must create the cache group before you can assign an interface to the group.

---

#### USING THE WEB MANAGEMENT INTERFACE

To assign interface 3 to cache group 1:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to System in the tree view to expand the list of system configuration links.
4. Select the Port link from the menu. Port configuration information for all the ports is displayed.
5. Select the Modify button next to the entry for interface 3.

---

**NOTE:** Some of the fields and options on this panel apply to Layer 2 features. See the *Foundry Switch and Router Installation and Basic Configuration Guide* for information about these features.

---

6. Select 1 from the Cache Group ID pulldown menu.
7. Select the Apply button to assign the interface to the cache group.

---

**NOTE:** You must create the cache group before you can assign an interface to that group.

---

8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Defining Distribution of Web Requests Within a Cache Group

To define how requests are distributed among multiple web cache servers within a cache group, you can use the **hash-mask** <destination-ip-mask> <source-ip-mask> CLI command at the transparent cache level. By default, the destination IP mask is 255.255.255.0, and the source IP mask is 0.0.0.0. The ServerIron uses the source and destination IP addresses as hash values for all types of traffic except HTTP and SSL. For these two types of traffic, the ServerIron also uses the source and destination TCP ports in addition to the source and destination IP addresses as hash values.

The hash mechanism minimizes duplication of content on the cache servers by ensuring that a particular web site is always cached on the same cache server.

---

**NOTE:** If you configure the ServerIron for Server Load Balancing (SLB) in addition to TCS, and the SLB configuration provides load balancing for your cache servers, then content will be duplicated on the cache servers as a result of the SLB predictor (load balancing metric). The SLB predictor works differently from the TCS hash mechanism and assumes that content is duplicated across the load-balanced servers. For information about the SLB predictor, see "Load Balancing Method (Predictor)" on page 6-24.

---



---

**NOTE:** Traffic controlled by policy-based caching on an individual server level is load balanced, whereas traffic for the other cache servers is partitioned according to the hash feature. See "Policy-Based Caching" on page 10-37.

---



---

**NOTE:** If you use Content Aware Cache Switching (URL switching in a TCS environment), URL string hashing is used to select a cache server within a server group. Content duplication is minimized because requests for cached content always go to the same cache server. See "Content Aware Cache Switching" on page 10-45 and "URL String Hashing" on page 11-35 for more information.

---

## Distribution Algorithm

When a cache group contains multiple cache servers, the ServerIron distributes traffic across the caches. The ServerIron distributes the traffic using a hashing feature. The hashing feature uses a source hash mask and a destination hash mask for each cache group. The ServerIron maintains a separate hash table for each cache group.

The masks determine how much of the source and destination IP addresses are used by the hash function. The ServerIron uses the hash masks to select a cache server. The ServerIron uses the following hash masks by default:

- Destination Hash Mask: 255.255.255.0

- Source Hash Mask: 0.0.0.0

In the default hash mask, the first three octets of the destination address are significant and the source address is not significant. Therefore, traffic addressed to any of the addresses in a Class-C subnet always goes to the same cache server, regardless of the source address.

Foundry devices use the following algorithm for distributing traffic among the cache servers:

- "AND" the destination IP address and destination IP mask to get d1.d2.d3.d4.
- "AND" the source IP address and source IP mask to get s1.s2.s3.s4.
- Add the 8-byte values d1, d2, d3, d4, s1, s2, s3 and s4 to get a 1-byte hash value.
- Use a 1-byte hash value to map to an entry in the hash table. Each entry maps to an active cache server.

The ServerIron contains 256 hash slots. If you do not assign weights to the cache servers (see "Modifying the Weight of a Server" on page 10-18), the software divides the hash slots evenly among the cache servers. If you assign differing weights to the cache servers, the software assigns hash slots to the cache servers based on the ratios of their relative weights.

The hashing feature allows the switch to spread the traffic across the caches and minimizes duplicate data on the cache servers.

If all the cache servers become unavailable, traffic flows across the switch at Layer 2 and users go directly out to the Internet. The ServerIron does not drop the traffic.

Table 10.2 on page 10-13 shows other examples of how the hash masks work.

**Table 10.2: Example TCS Hash Masks**

Destination Mask	Source Mask	Destination IP Address	Source IP Address	Cache Server
255.255.255.0	0.0.0.0	125.24.32.12	Any	C1
		125.24.32.210	Any	C1
		125.24.33.210	Any	C2
		125.24.34.210	Any	C3
255.255.255.192	0.0.0.0	125.24.32.12	Any	C1
		125.24.32.70	Any	C2
		125.24.32.190	Any	C3
255.255.255.0	0.0.0.255	125.24.32.12	149.165.16.233	C1
		125.24.32.12	189.12.122.233	C1
		125.24.32.12	189.12.122.200	C2

#### *USING THE CLI*

To direct all web queries destined for the same web site (such as "www.rumors.com") to the same cache server for processing, enter the following **hash-mask** command:

```
ServerIron(config-tc-1)# hash-mask 255.255.255.255 0.0.0.0
```

**NOTE:** This is useful for networks that have many users accessing the same web site locations. It may be more useful to use only the first three octets of the Destination IP address (255.255.255.0) for web sites that may return multiple web server addresses (for example "www.rumors1.com" and "www.rumors2.com") in response to www.rumors.com queries.

---

To direct all users from the same Class B sub-net (255.255.0.0) to either server1 or server2 and to direct all redundant requests destined to the same web site (255.255.255.0) to the same web cache server, enter the following **hash-mask** command:

```
ServerIron(config-tc-1)# hash-mask 255.255.255.0 255.255.0.0
```

**Syntax:** hash-mask <destination-mask> <source-mask>

#### [USING THE WEB MANAGEMENT INTERFACE](#)

You cannot configure this option using the Web management interface.

### **Adding a Virtual IP Address for Policy-Based Cache Failover (CFO)**

If the RAS connecting the clients to the ServerIron uses policy filters to forward client requests to a virtual IP address, you need to enable policy-based Cache Failover (CFO) and add the virtual IP address that the router's policy uses to the cache group. In this type of configuration, CFO prevents client requests from becoming lost in a "black hole" when the cache servers are unavailable. When you configure the ServerIron for CFO, the ServerIron forwards such requests back to the RAS for forwarding to the Internet. Thus, clients still receive the requested content even though the cache servers are unavailable.

To configure CFO, make sure you do the following:

1. Set up the router and aim the policy on the router at the virtual address on the ServerIron rather than at the address of the cache.
2. Define the cache or caches on the ServerIron and place them into cache group 1.
3. Define the virtual IP address in cache group 1.
4. Define the IP cache policy as a global cache.

---

**NOTE:** For CFO, you must define a global policy, not a local policy.

---

See "Policy-Based Cache Failover (CFO)" on page 10-41 for an example of this type of configuration.

#### [USING THE CLI](#)

```
ServerIron(config-tc-1)# virtual-ip 209.157.22.77
```

**Syntax:** virtual-ip <ip addr>

#### [USING THE WEB MANAGEMENT INTERFACE](#)

You cannot add a virtual IP address to a cache group using the Web management interface.

### **Cache Server Spoofing**

In TCS, when a client makes a request for HTTP content on the Internet, the ServerIron directs the request to a cache server, rather than to the Internet. If the requested content is not on a cache server, it is obtained from an origin Web server on the Internet, stored on a cache server to accommodate future requests, and sent from the cache server back to the requesting client.

---

**NOTE:** You cannot use the cache server spoofing feature with the Reverse Proxy SLB feature on the same ServerIron.

---

When a cache server makes a request for content from the origin server, it can do one of the following:

- The cache server replaces the requesting client's IP address with its own before sending the request to the Internet. The origin server then sends the content to the cache server. The cache server stores the content



and sends it to the requesting client, changing the source IP address from its own to the origin server's IP address.

- The cache server does not replace the requesting client's IP address with its own. Instead, the cache server sends the request to the Internet using the requesting client's IP address as the source. This allows the origin server to perform authentication and accounting based on the client's IP address, rather than the cache server's IP address. This functionality is known as **cache server spoofing**.

When cache server spoofing support is enabled, the ServerIron does the following with requests sent from a cache server to the Internet:

1. The ServerIron looks at the MAC address to see if the packet is from a cache server. Note that the ServerIron and the cache server cannot be separated by any router hops; they must be on the same physical segment. The ServerIron uses an ARP request to get the MAC address of each configured cache server.
2. If the MAC address indicates that the packet is from a cache server, the ServerIron checks the source IP address. If the source IP address does not match the cache server's IP address, the ServerIron concludes that this is a spoofed packet.
3. The ServerIron creates a session entry for the source and destination (IP address, port) combination, and then sends the request to the Internet.

When the origin server sends the content back, the ServerIron looks for a session entry that matches the packet. If the session entry is found, the ServerIron sends the packet to the appropriate cache server.

To enable cache server spoofing support:

```
ServerIron(config)# server cache-group 1
ServerIron(config-tc-1)# spoof-support
```

**Syntax:** [no] spoof-support

The **no** form of the command disables cache server spoofing support. Cache server spoofing support is disabled by default.

The **show cache-group** command displays the number of spoofed packets encountered by the ServerIron. For example:

```
ServerIron# show cache-group
Cache-group 1 has 1 members Admin-status = Enabled Active = 0
Hash_info: Dest_mask = 255.255.255.0 Src_mask = 0.0.0.0

Cache Server Name          Admin-status Hash-distribution
cs-1                        1              0

HTTP Traffic  From <-> to  Web-Caches

Name: cs-1                IP: 1.2.5.3                State: 1    Groups = 1

                State  CurConn  TotConn  Host->Web-cache  Web-cache->Host
                State  CurConn  TotConn  Spoof pkts  Spoof octs  Spoof pkts  Spoof octs
Client          enabled 0        0        0            0          0            0
Web-Server      active 0        0        0            0          0            0
Total           0        0        0            0          0            0
```

**Syntax:** show cache-group

---

**NOTE:** Information on spoofed packets is displayed only if cache server spoofing support is enabled.

---

## Modifying Global TCS Parameters

You can modify the following global parameters:

- Cache Route Optimization
- Force shutdown

**NOTE:** For information about modifying ping and HTTP keepalive parameters, see “Configuring Port and Health Check Parameters” on page 12-1. For information about adding source IP addresses, see “Source IP Address” on page 6-30.

### Enabling or Disabling Cache Route Optimization

Cache Route Optimization (CRO) is useful for situations in which a cache server’s default gateway is the Border Access Router (BAR) that goes to the Internet, instead of the remote access server (RAS) that goes to the HTTP clients. When you enable CRO, the ServerIron intelligently sends cache responses directly to the RAS at Layer 2 instead of sending them to the BAR for switching back through the ServerIron to the RAS.

CRO is described in detail in “Cache Route Optimization (CRO)” on page 10-35.

CRO is disabled by default. Use one of the following methods to enable the feature.

#### USING THE CLI

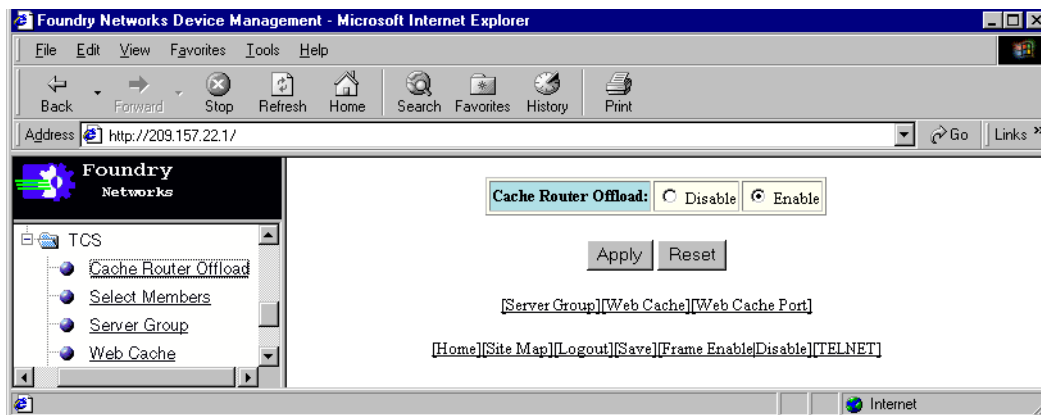
```
ServerIron(config)# server cache-router-offload
```

**Syntax:** [no] server cache-router-offload

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to TCS in the tree view to expand the list of Transparent Cache Switching option links.
4. Select the Server Group link from the menu.
5. Select Cache Router Offload to display the panel shown in Figure 10.7.

**Figure 10.7** Cache Router Offload panel



6. Select Enable.
7. Select Apply to implement the change.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device’s flash memory.

## Enabling or Disabling Force Shutdown

SLB and TCS allow the graceful shutdown of servers and services. By default, when a service is disabled or deleted, the ServerIron does not send new connections the real servers for that service. However, the ServerIron does allow existing connections to complete normally, however long that may take.

You can use the force shutdown option (sometimes called the force delete option) to force the existing connections to be terminated within two minutes.

---

**NOTE:** If you disable or delete a service, do not enter an additional command to reverse the command you used to disable or delete the service, while the server is in graceful shutdown.

---

---

**NOTE:** See “Shutting Down a Cache Server” on page 10-31 for important information about shutting down services or servers.

---

### USING THE CLI

Suppose you have unbound the Telnet service on real server 15 but you do not want to wait until the service comes down naturally. You can use the **force-delete** command to force TCS connections to be terminated:

```
ServerIron(config)# server force-delete
```

**Syntax:** server force-delete

### USING THE WEB MANAGEMENT INTERFACE

To enable or disable force shutdown:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu.
5. Select Force Shutdown.
6. Select the Apply button to assign the change.

---

**NOTE:** Although you enable or disable force shutdown from an SLB panel, the parameter setting applies globally to both SLB and TCS.

---

7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Modifying Default Settings for Cache Servers

You can modify the following default settings for cache servers in the network:

- Maximum connections for a server
- Weight of a server
- FastCache
- Destination NAT
- Source NAT
- Remote Cache
- Policy-based caching

---

**NOTE:** You also can configure Layer HTTP health checks for the cache servers. See “Layer 7 Health Checks” on page 12-3.

---

## Modifying Maximum Connections for a Cache Server

You can limit the maximum number of connections supported on a server-by-server basis. By setting a limit, you can avoid a condition where the capacity threshold of a cache server is exceeded.

When a cache server reaches the maximum defined connection threshold, the ServerIron sends an SNMP trap. When all the cache servers in a cache group reach their maximum connection threshold, the ServerIron sends client requests to the Internet.

Up to 1,000,000 sessions are supported. This is the default.

### USING THE CLI

To limit the connections to a maximum of 100,000 for cache server1 and 200,000 for server2 in the network seen in Figure 10.1, enter the following commands:

```
ServerIron(config)# server cache-name server1
ServerIron(config-rs-server1)# max-conn 100000
ServerIron(config-rs-server1)# server cache-name server2
ServerIron(config-rs-server2)# max-conn 200000
ServerIron(config-rs-server2)# end
ServerIron# write mem
```

**Syntax:** max-conn <32768-1000000> for 32M systems; max-conn <32768-1600000> for 8M systems

---

**NOTE:** The **max-conn** command is in the Real Server Level of the CLI along with the port and weight commands described in the next two sections.

---

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to TCS in the tree view to expand the list of Transparent Cache Switching option links.
4. Select the Web Cache link from the TCS configuration sheet.
5. Select the Modify button next to the server to be modified. The Web Cache panel will appear.
6. Enter a value between 32,768 and 1,000,000 in the Maximum Connections field. Enter a value between 32,768 and 160,000 if configuring an 8M switch.
7. Select the Modify button to assign the changes.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Modifying the Weight of a Server

This parameter allows you to assign a performance weight to each server. Servers assigned a larger or higher weight receive a larger percentage of connections. Possible values are 1 – 20 with a default value of 1.

### USING THE CLI

To set the weight for cache server1 to 5 from the default value of 1, enter the following commands:

```
ServerIron(config)# server cache-name server1
ServerIron(config-rs-server1)# weight 5
```

**Syntax:** weight <1-20>

### USING THE WEB MANAGEMENT INTERFACE

To modify the weight assigned to a cache server:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to TCS in the tree view to expand the list of Transparent Cache Switching option links.
4. Select the Web Cache link from the TCS configuration sheet.
5. Select the Modify button next to the server to be modified. The Web Cache panel will appear.
6. Enter a value from 1 – 20 in the Weight field.
7. Select the Modify button to assign the changes.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Enabling FastCache

By default, the ServerIron uses cache responses to client requests as a means to assess the health of the cache server. However, in an asymmetric topology where the cache server uses a path to the client that does not pass through the ServerIron, the ServerIron does not observe the return traffic. As a result, the ServerIron concludes that the cache server has failed even though the server might still be healthy.

When the ServerIron concludes that a cache server is unavailable, the ServerIron stops sending client requests to the cache server. You can override this behavior by enabling the FastCache feature. The FastCache feature configures the ServerIron to continue sending client requests to a cache server even if the ServerIron does not see responses from the server.

#### USING THE CLI

```
ServerIron(config)# server cache-name server1
ServerIron(config-rs-server1)# asymmetric
```

**Syntax:** asymmetric

#### USING THE WEB MANAGEMENT INTERFACE

To enable the FastCache feature for a cache server:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to TCS in the tree view to expand the list of Transparent Cache Switching option links.
4. Select the Web Cache link from the TCS configuration sheet.
5. Select the Modify button next to the server to be modified. The Web Cache panel will appear.
6. Select the Asymmetric field to mark the FastCache feature enabled.
7. Select the Modify button to assign the changes.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Enabling Destination NAT

By default, the ServerIron translates the destination MAC address of a client request into the MAC address of the cache server. However, the ServerIron does not translate the IP address of the request to the cache server's IP address. Instead, the ServerIron leaves the destination IP address untranslated.

This behavior assumes that the cache server is operating in promiscuous mode, which allows the cache server to receive requests for any IP address so long as the MAC address in the request is the cache server's. This behavior works well in most caching environments. However, if your cache server requires that the client traffic arrive in directed IP unicast packets, you can enable destination NAT.

Destination NAT is disabled by default.

---

**NOTE:** This option is rarely used. If your cache server operates in promiscuous mode, you probably do not need to enable destination NAT. Otherwise, enable destination NAT. Consult your cache server documentation if you are unsure whether you need to enable destination NAT.

---

#### *USING THE CLI*

```
ServerIron(config)# server cache-name server1
```

```
ServerIron(config-rs-server1)# dest-nat
```

**Syntax:** dest-nat

#### *USING THE WEB MANAGEMENT INTERFACE*

To enable destination NAT for a cache server:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to TCS in the tree view to expand the list of Transparent Cache Switching option links.
4. Select the Web Cache link from the TCS configuration sheet.
5. Select the Modify button next to the server to be modified. The Web Cache panel will appear.
6. Select the Destination NAT field to mark the feature enabled.
7. Select the Modify button to assign the changes.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

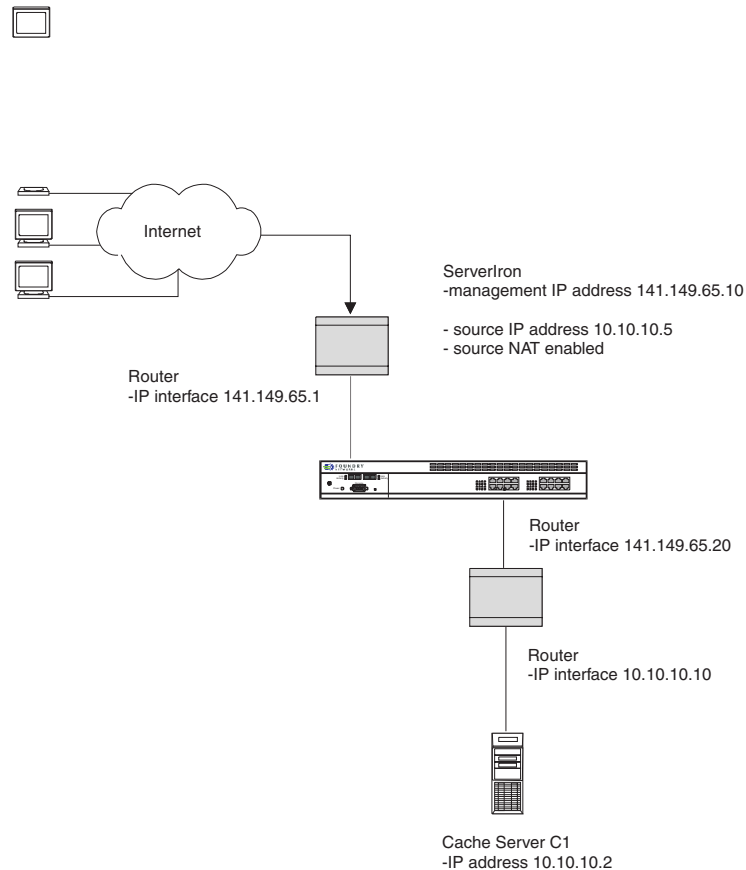
#### **Enabling Source NAT**

Normally, when the ServerIron redirects a client's web request to a cache server, the ServerIron translates the destination MAC address of a client request into the MAC address of the cache server. However, the ServerIron does not translate the source or destination IP addresses in the client's request.

Generally, in network topologies where the ServerIron and cache server are directly connected or connected through a Layer 2 switch or bridge, the cache's response to a client query always passes back through the ServerIron. The ServerIron uses the cache response to assess the health of the cache server. When the ServerIron passes a cache response to the client, the ServerIron assumes the cache server is healthy.

However, if the time since the last packet the ServerIron sent to the cache server and the cache server's response increases significantly, or the cache server's reply never reaches the ServerIron but instead takes an alternate path to the client, the ServerIron assumes that the cache server has stopped responding. When this occurs, the ServerIron marks the cache server FAILED and stops redirecting client queries to the cache server.

You can ensure that cache server replies always pass back through the ServerIron by configuring Source NAT.

**Figure 10.8 Using Source NAT with TCS**

In this example, the ServerIron and cache server are connected by a router and are in different sub-nets. In a topology where the cache server's response is guaranteed to pass back through the ServerIron, you may not need to configure Source NAT. However, if the cache server's reply can reach the client by a path that does not pass through the ServerIron, you need to configure Source NAT.

To configure Source NAT:

- Enable the Source NAT feature. You can enable the feature at the cache group level for all cache servers or at the cache server level for individual servers.
- Configure a source IP address. A source IP address allows the ServerIron to be a member of more than one sub-net. If the cache server and ServerIron are in different sub-nets, configure a source IP address that is in the cache server's sub-net.

#### *USING THE CLI*

To enable Source NAT globally for all cache servers and configure a source IP address, enter commands such as the following:

```
ServerIron(config)# server source-ip 10.10.10.5
ServerIron(config)# server cache-group 1
ServerIron(config-tc-1)# source-nat
ServerIron(config-tc-1)# dest-nat
```

These commands configure a source IP address at the global CONFIG level of the CLI, then change the CLI to the cache group configuration level and enable source NAT and Destination NAT. Source NAT configures the ServerIron to change the source IP address in a client query from the client's IP address to configured source IP address. Destination NAT configures the ServerIron to change the destination IP address of the client's request to the IP address of the cache server.

**Syntax:** [no] source-ip <ip-addr> <network-mask> <default-gateway>

---

**NOTE:** The gateway parameter is required. If you do not want to specify a gateway, enter "0.0.0.0".

---

**Syntax:** [no] source-nat

To enable source NAT on a specific cache server instead of at the cache group configuration level for all cache servers, enter commands such as the following:

```
ServerIron(config)# server cache-name C1
ServerIron(config-rs-C1)# source-nat
ServerIron(config-rs-C1)# dest-nat
```

The commands in this example enable Source NAT and Destination NAT on cache server C1 only. This example assumes that the source IP address also is configured as shown in the previous example.

### Enabling Remote Cache

The configuration examples in “Enabling Source NAT” on page 10-20 assume that Proxy ARP is enabled on the router that connects the ServerIron to the cache servers. When Proxy ARP is enabled on the router, the router informs the ServerIron that the router can respond on behalf of the cache server. The ServerIron uses ARP requests as part of the keepalive health checking mechanism, so Proxy ARP enables the keepalive health checking mechanism to function.

If Proxy ARP is disabled on the router, the keepalive health checking mechanism believes the cache server cannot be reached and does not mark the server ACTIVE or direct request to the cache server.

You can enable the ServerIron to overcome the limitation posed by the absence of Proxy ARP by enabling the Remote Cache feature for the cache server.

To enable the Remote Cache feature, enter a command such as the following:

```
ServerIron(config)# server cache-name C1
ServerIron(config-rs-C1)# remote-cache
```

**Syntax:** [no] remote-cache

This example enables Remote Cache on cache server C1. Since the feature is enabled, the ServerIron can successfully perform health checks on the cache server despite the fact that Proxy ARP is disabled on the router that connects the ServerIron to the cache server.

This example assumes that a source IP address is configured and Source NAT and Destination NAT also are enabled, if applicable.

### Enabling Policy-Based Cache Switching for an Individual Server

By default, the ServerIron uses the hash distribution algorithm described in “Defining Distribution of Web Requests Within a Cache Group” on page 10-12 to balance the cache load across the cache servers within a cache group. The content that is cached on a server depends upon the hash algorithm.

However, you might want to control the content that is cached by specific cache servers. For example, if you have a cache server that is preconfigured with specific web sites, you can configure the ServerIron to cache updates to those sites only on the preconfigured cache server, not on other cache servers. For a detailed example, see “Policy-Based Caching” on page 10-37.

---

**NOTE:** Traffic controlled by policy-based caching on an individual server level is load balanced, whereas traffic for the other cache servers is partitioned according to the hash feature.

---

### USING THE CLI

```
ServerIron(config)# server cache-name server1
ServerIron(config-rs-server1)# filter-match
```

**Syntax:** filter-match



### USING THE WEB MANAGEMENT INTERFACE

To enable policy-based caching for a cache server:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to TCS in the tree view to expand the list of Transparent Cache Switching option links.
4. Select the Web Cache link from the TCS configuration sheet.
5. Select the Modify button next to the server to be modified. The Web Cache panel will appear.
6. Select the Filter Match field to mark the feature enabled.
7. Select the Modify button to assign the changes.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Modifying Default TCP/UDP Port Parameters

You can modify the following parameters for individual TCP/UDP ports:

- Port priority
- Health check state and Layer 7 parameters (see "Customizing Layer 7 Health Checks" on page 12-31)
- Maximum connection rate

### Modifying Port Priority

Port priority allows you to assign a higher priority to a specific TCP/UDP port on a specific physical port. Assigning a higher priority will result in that TCP/UDP port receiving preference over others active on the physical port with lower priorities assigned.

You can define a number of profiles that can be assigned to individual physical ports. For example, you can configure one priority profile (ID 1) that assigns a higher priority to HTTP traffic and another that assigns a higher priority to FTP traffic (ID 2). You can then assign these priority profiles to physical ports. This results in all traffic defined by that profile receiving priority over other types of traffic operating on that port.

### USING THE CLI

```
ServerIron(config-if-1)# priority cache
```

**Syntax:** priority normal | high | cache

### USING THE WEB MANAGEMENT INTERFACE

To modify the TCP/UDP priority on a port:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to System in the tree view to expand the list of system configuration links.
4. Select the Layer 4 QoS link. The panel shown in Figure 10.2 will appear.
5. Enter a value from 1 – 1024 into the ID field.
6. Select Port as the scope of the priority. Notice that the TCP/UDP port priority can also be assigned on a global basis (switch).
7. Select Cache as the QoS type.
8. Select either TCP or UDP protocol.
9. Select the TCP/UDP port that is being defined in the priority profile from the pulldown menu.
10. Select the Add button to assign the changes.

11. Select the [Port QoS](#) link to display the panel shown in Figure 10.3 on page 10-7.
12. Select the port that the TCP/UDP priority is to be assigned.
13. Enter the number or numbers of the priority profile(s) that are to be assigned to the port in the QoS ID list.
14. Select the Add button to assign the priority to the port.
15. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Connection Rate Limiting

Connection Rate Limiting (CRL) enables you to limit the connection rate to a cache server. The ServerIron limits the number of new port connections per second to the number you specify.

The ServerIron increments the connection counter for cache connections only after the ServerIron selects one for the connection. If the ServerIron cannot serve a client request because a cache already has the maximum number of connections for the current second for the requested port, the ServerIron tries another cache. If there are no caches available, the ServerIron directs the request to the Internet.

If you configure a limit for TCP and also for an individual application port, the ServerIron uses the lower limit. For example, if you limit new TCP connections to a real server to 1000 per second and also limit new HTTP connections to 600 per second, the ServerIron limits connections to TCP port HTTP to 600 per second.

---

**NOTE:** The ServerIron counts only the new connections that remain in effect at the end of the one second interval. If a connection is opened and terminated within the interval, the ServerIron does not include the connection in the total for the server.

---



---

**NOTE:** The connection limit you specify is enforced on an individual WSM CPU basis. Thus, each WSM CPU allows up to the number of connections you specify. For example, if you specify a maximum TCP connection rate of 800 connections per second, each WSM CPU allows up to 800 TCP connections per second, for a total of 2400 TCP connections per second.

---

To limit the number of new TCP connections a cache can receive each second, enter commands such as the following:

```
ServerIron(config)# server cache C1 5.6.7.8
ServerIron(config-rs-C1)# max-tcp-conn-rate 2000
```

You also can specify the connection rate for an individual port. Here is an example:

```
ServerIron(config)# server cache C1 5.6.7.8
ServerIron(config-rs-C1)# port http
ServerIron(config-rs-C1)# port http max-tcp-conn-rate 2000
```

**Syntax:** max-tcp-conn-rate <num>

The <num> parameter specifies the maximum number of connections per second. There is no default.

**Syntax:** port <TCP/UDP-portnum> max-tcp-conn-rate <num>

The **port** <TCP/UDP-portnum> parameter specifies the application port.

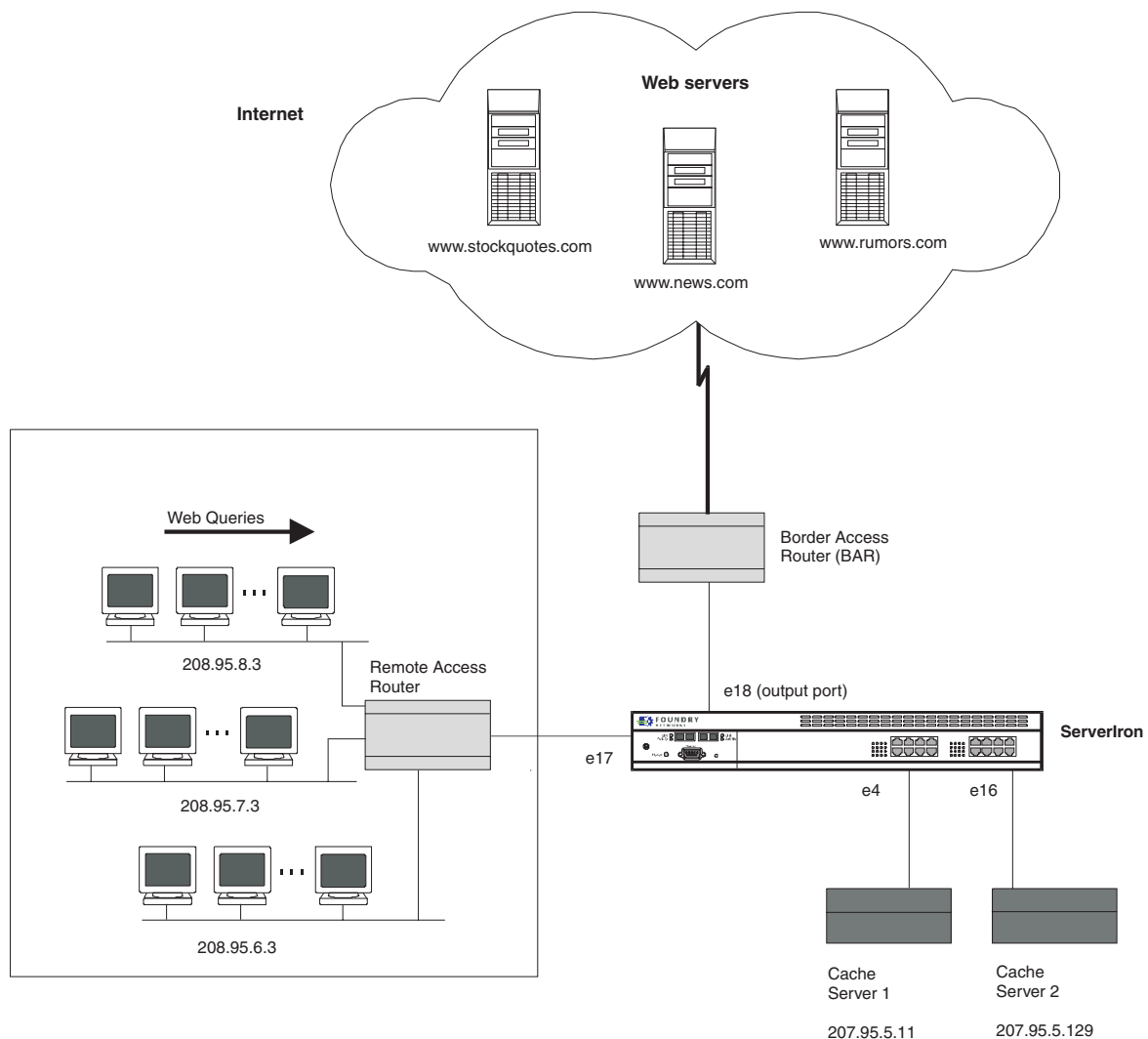
The <num> parameter specifies the maximum number of connections per second.

### Using IP Filters to Control Caching

You can turn off web caching for a certain range of source or destination addresses to allow filtering on an address basis using IP filters.

There are two choices for the filter actions:

- Permit – The ServerIron redirects the user request to a cache server.
- Deny – The ServerIron does not redirect the user request to a cache server but instead passes the request to the Internet.

**Figure 10.9 Using IP filters to bias traffic away from cache servers**

For the example in Figure 10.9, you can use a deny filter to force all web queries from a specific sub-net to go directly to the Internet (the normal destination) rather than be redirected to cache servers.

If you want all web queries from sub-net 208.95.6.0 to go directly to the Internet, rather than be redirected to the cache servers, define a deny and permit filter as shown below.

#### USING THE CLI

```
ServerIron(config)# ip filter 1 deny 208.95.6.0 255.255.255.0 any
ServerIron(config)# ip filter 1024 permit any any
```

**NOTE:** When defining filters, you must define both a deny filter and a permit filter.

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to System in the tree view to expand the list of system configuration links.

4. Select the IP Filter link. The panel shown in Figure 10.10 will appear.

**Figure 10.10** Defining an IP filter

The screenshot shows the 'Foundry Networks Device Management' web interface in Microsoft Internet Explorer. The address bar shows 'http://209.157.22.1/'. The left navigation pane shows a tree structure with 'ServerIron' expanded, and 'IP Filter' selected under the 'Configure' section. The main content area is titled 'IP Filter' and contains the following fields:

- ID:** 1
- Action:** ☒ Deny ☐ Permit
- Source Address:** 208.95.6.3
- Source Mask:** 255.255.255.0
- Destination Address:** 0.0.0.0
- Destination Mask:** 0.0.0.0
- Protocol:** TCP
- Operator:** Equal
- TCP/UDP port:** 80 ☐ Filter Established TCP

At the bottom of the form are buttons for 'Add', 'Modify', 'Delete', and 'Reset'. Below these buttons is a '[Show]' link. At the very bottom of the page are links for '[Home]', '[Site Map]', '[Logout]', '[Save]', '[Frame Enable]', '[Disable]', and '[TELNET]'.

5. Select the type of filter (action) to be defined: deny or permit.
6. If you want to define the filter for a specific source address, enter the IP address of the source sub-net in the Source Address field. If you leave the default address 0.0.0.0 in this field, the filter is applied to all received traffic.
7. If you entered an IP address in the previous step, enter the source mask in the Source Mask field.
8. If you want to define the filter for a specific destination address, enter the IP address of the destination sub-net in the Destination Address field. If you leave the default address 0.0.0.0 in this field, the filter is applied to all forwarded traffic.
9. If you entered an IP address in the previous step, enter the destination mask in the Destination Mask field.
10. Enter the protocol to be filtered. In the case of TCS, the only protocol you can filter is TCP.
11. Select the comparison operator from the Operator field.
12. Enter "80" (the well-known port number for HTTP) or another HTTP port number if applicable.
13. Select the Add button to assign the change.

**NOTE:** You do not need to specify an operator or TCP/UDP port number for TCS filters. Leave the default values (Equal and 0) in the Operator and TCP/UDP port fields.

14. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Displaying Configuration Information and Statistics

You can display configuration information and statistics for cache servers using either of the following methods.

### USING THE CLI

To display cache information, enter the following command at any level of the CLI:

```
ServerIron# show cache-group
Cache-group 1 has 1 members Admin-status = Enabled
Hash_info: Dest_mask = 255.255.255.0 Src_mask = 0.0.0.0

      Cache Server Name                Admin-st  Hash-distribution
      cf                             6         4

HTTP Traffic  From <-> to  Web-Caches
=====

Name: cf                IP: 209.157.23.195    State: 6    Groups = 1

      State CurConn TotConn      Host->Web-cache      Web-cache->Host
      Client active    0  386581  Packets  Octets  Packets  Octets
Web-Server Active    0    0      0      0      0      0
Total          0  386581  1932917 185657048 1547981 393357745

HTTP Uncached traffic
=====

Traffic to Web-server port 1

      Client->Web-Server      Web-Server->Client
      Client-port      Packets  Octets  Packets  Octets
2              8230    670375    8038    7348299
4              97      8129     92      83257
Total          8327    678504    8130    7431556
```

**Syntax:** show cache-group

This display shows the following information.

**Table 10.3: TCS Information**

This Field...	Displays...
<b>Global cache group information</b>	
This section of the display lists global information for the cache group.	
Admin-status	<p>The administrative status of the cache group. The status can be one of the following:</p> <ul style="list-style-type: none"> <li>• Disabled</li> <li>• Enabled</li> </ul>

**Table 10.3: TCS Information (Continued)**

This Field...	Displays...
Hash_info	<p>The source and destination mask for the cache distribution hash value. The ServerIron compares the web site's IP address to the hash mask to determine which cache server to send a request for a given web site to.</p> <p>As long as the cache server is available, the ServerIron always sends requests for a given IP address to the same cache. If a cache becomes unavailable, the ServerIron directs requests for web sites normally served by that cache to the remaining cache servers until the unavailable cache server becomes available again.</p>
Cache Server Name	The names of the cache servers in the cache group. These are the names you assigned when you configured cache server information on the ServerIron.
Admin-st	<p>The administrative state of the cache server, which can be one of the following:</p> <ul style="list-style-type: none"> <li>• 1 – Enabled</li> <li>• 2 – Failed</li> <li>• 3 – Testing</li> <li>• 4 – Suspect</li> <li>• 5 – Graceful shutdown</li> <li>• 6 – Active</li> </ul>
Hash-distribution	The number of hash distribution slots used by the cache server. The ServerIron has 256 hash distribution slots available. A hash distribution slot associates a web site destination IP addresses with a specific cache server. See "Defining Distribution of Web Requests Within a Cache Group" on page 10-12 for more information about has values.
<b>Traffic statistics for traffic between clients and the cache</b>	
Name	The cache server name
IP	The cache server's IP address
State	<p>The administrative state of the cache server, which can be one of the following:</p> <ul style="list-style-type: none"> <li>• 1 – Enabled</li> <li>• 2 – Failed</li> <li>• 3 – Testing</li> <li>• 4 – Suspect</li> <li>• 5 – Graceful shutdown</li> <li>• 6 – Active</li> </ul>
Groups	The cache groups of which the cache server is a member.

**Table 10.3: TCS Information (Continued)**

This Field...	Displays...
State	The state of the server, which can one of the following: <ul style="list-style-type: none"> <li>• 1 – Enabled</li> <li>• 2 – Failed</li> <li>• 3 – Testing</li> <li>• 4 – Suspect</li> <li>• 5 – Graceful shutdown</li> <li>• 6 – Active</li> </ul>
CurConn	The number of currently active connections between hosts and the cache server.
TotConn	The total number of connections between hosts and the cache server.
Host->Web-cache packets	<ul style="list-style-type: none"> <li>• For the Client row – the total number of packets from clients to the cache server</li> <li>• For the Web-Server row – the total number of packets from the cache server to web servers</li> </ul>
Host->Web-cache octets	<ul style="list-style-type: none"> <li>• For the Client row – the total number of octets from clients to the cache server</li> <li>• For the Web-Server row – the total number of octets from the cache server to web servers</li> </ul>
Web-cache->Host packets	<ul style="list-style-type: none"> <li>• For the Client row – the total number of packets from the cache server to clients</li> <li>• For the Web-Server row – the total number of packets from the web servers to the cache server</li> </ul>
Web-cache->Host octets	<ul style="list-style-type: none"> <li>• For the Client row – the total number of octets from the cache server to clients</li> <li>• For the Web-Server row – the total number of octets from the web servers to the cache server</li> </ul>

**Traffic statistics for traffic between clients and web servers**

The statistics for this section are for traffic that did not go to the cache server. Generally, statistics in this section accumulate when the cache server is not available. When the cache server is not available, the ServerIron sends client requests directly to the network or Internet for servicing by the web servers.

Traffic to Web-server port	The ServerIron port attached to the web servers. Generally, this is the port attached to the Border Access Router (BAR) that goes to the rest of the network or to the Internet.
Client Port	For each row of statistics, the ServerIron port attached to clients. In the example above, rows of statistics are displayed for ServerIron ports 2 and 4, each of which are attached to clients.
Client->Web-Server Packets	The total number of packets from clients that the ServerIron has sent directly to the web servers because the cache server was unavailable.
Client->Web-Server Octets	The total number of octets from clients that the ServerIron has sent directly to the web servers because the cache server was unavailable.

**Table 10.3: TCS Information (Continued)**

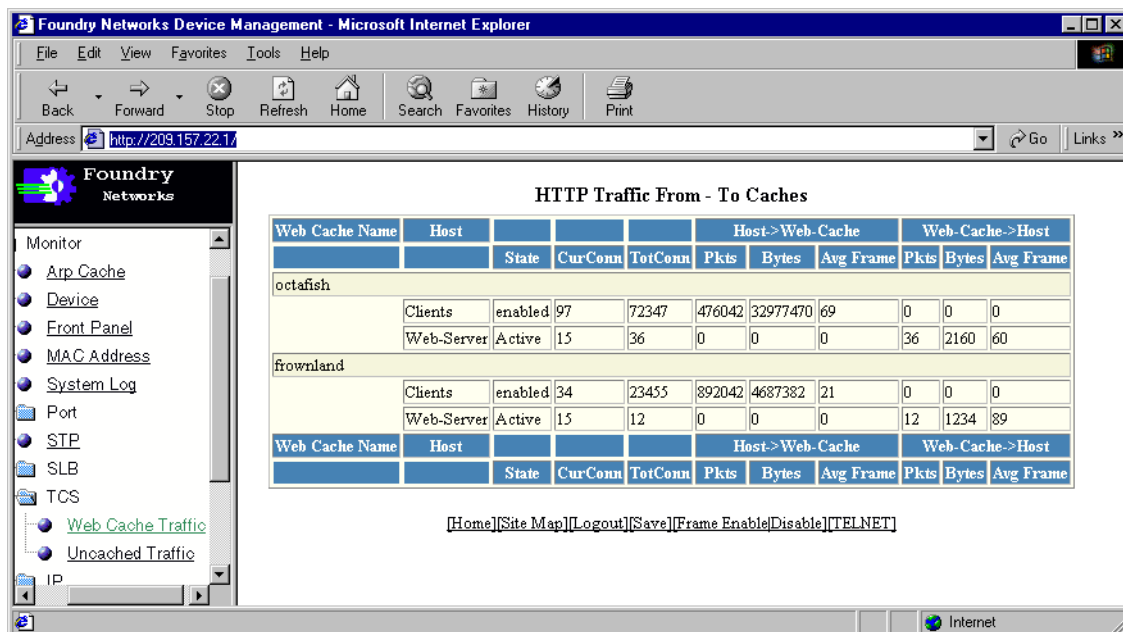
This Field...	Displays...
Web-Server->Client Packets	The total number of packets from web servers to clients.
Web-Server->Client Octets	The total number of octets from web servers to clients.
Total	Cumulative totals for uncached traffic for all client ports.

### USING THE WEB MANAGEMENT INTERFACE

To view configuration details for TCS:

1. Log on to the device using a valid user name and password for read access.
2. Click on the plus sign next to Monitor in the tree view to expand the list of system monitoring options.
3. Click on the plus sign next to TCS in the tree view to expand the list of TCS monitoring links.
4. Select the Web Cache Traffic link to view a traffic summary for cache servers defined on the ServerIron as shown in Figure 10.11.

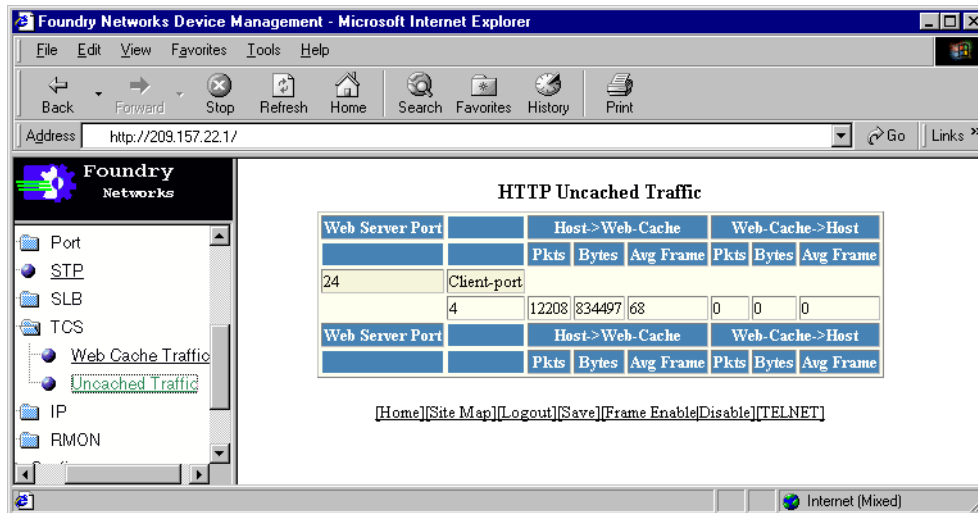
**Figure 10.11 Show web cache traffic display**



Select the Uncached Traffic link to view a traffic summary for that traffic that is being sent directly to the Internet.



Figure 10.12 Show uncached traffic display



For information about the fields, see the descriptions for the CLI **show cache-group** command above.

## Shutting Down a Cache Server

The force shutdown feature (sometimes called the force delete feature) allows you to force termination of existing SLB connections. This feature assumes that you already have shut down a TCP/UDP service on the real server or you have shut down the real server itself.

There are several methods for shutting down a cache server. Some methods involve changing the ServerIron configuration while other methods involve shutting down the cache server itself. Each method has consequences, so choose the method that works best in your situation.

- Edit the cache server configuration on the ServerIron to disable the HTTP (or other) port on the server. For example, to disable port 80 (HTTP), you can use the **port http disable** command at the cache level of the CLI. If you use this method, you do not need to re-define the cache server to add the server back to TCS. However, you do need to re-enable the disabled TCP/UDP ports.

Although the HTTP port is disabled in the ServerIron definition of the cache server, all the sites mapped to the cache server before the port was disabled remain mapped to the cache server. When the cache server comes back up, it gets the same traffic it used to have. While the cache server is disabled, the remaining cache server(s) temporarily handle caching for the down cache server's sites, but stop when the cache is restored. This behavior is the same as if the cache actually died.

---

**NOTE:** You might need to set the maximum connections parameter for the remaining cache servers, especially if the servers already run at a high percentage of their capacity when all cache servers are available. See "Modifying Maximum Connections for a Cache Server" on page 10-18.

---

- Delete the cache server from the ServerIron. This option immediately prevents new connections. The ServerIron ends existing connections after two minutes or, if you have enabled the force shutdown option, immediately.

Do not use this method unless you have only one cache server. If you use this method, to re-add the cache server to the ServerIron, you must redefine the cache server and re-assign it to a cache group. Moreover, because the ServerIron uses a hashing function to allocate contents among cache servers, the ServerIron allocates traffic to the remaining caches. If the deleted cache server is down for a while in a busy network, the traffic might be unevenly balanced between the cache server that was down and the other cache servers. To reset the hash function and thus rebalance the serving load, you need to reboot the ServerIron.

- Shut down the cache server itself, rather than change definitions on the ServerIron. When the cache server stops responding to health checks, the ServerIron removes the server from TCS. If you have only one cache server, user traffic is switched at Layer 2 to the Internet until the cache server comes back. If you have more than one cache server, the remaining cache servers provide service until the disabled cache server comes back.

This option is simple because it does not require any configuration changes on the ServerIron. However, this option immediately disconnects all users from the cache server, whereas the above options allow the server or service to gracefully shut down (unless you use the force shutdown option).

---

**NOTE:** You might need to set the maximum connections parameter for the remaining cache servers, especially if the servers already run at a high percentage of their capacity when all cache servers are available. See “Modifying Maximum Connections for a Cache Server” on page 10-18.

---

- To halt all caching activity, you can use one of the methods listed above for each cache server. Alternatively, you can remove the IP filter that controls caching.

## TCS Application Examples

The examples in this section illustrate implementations of the following features.

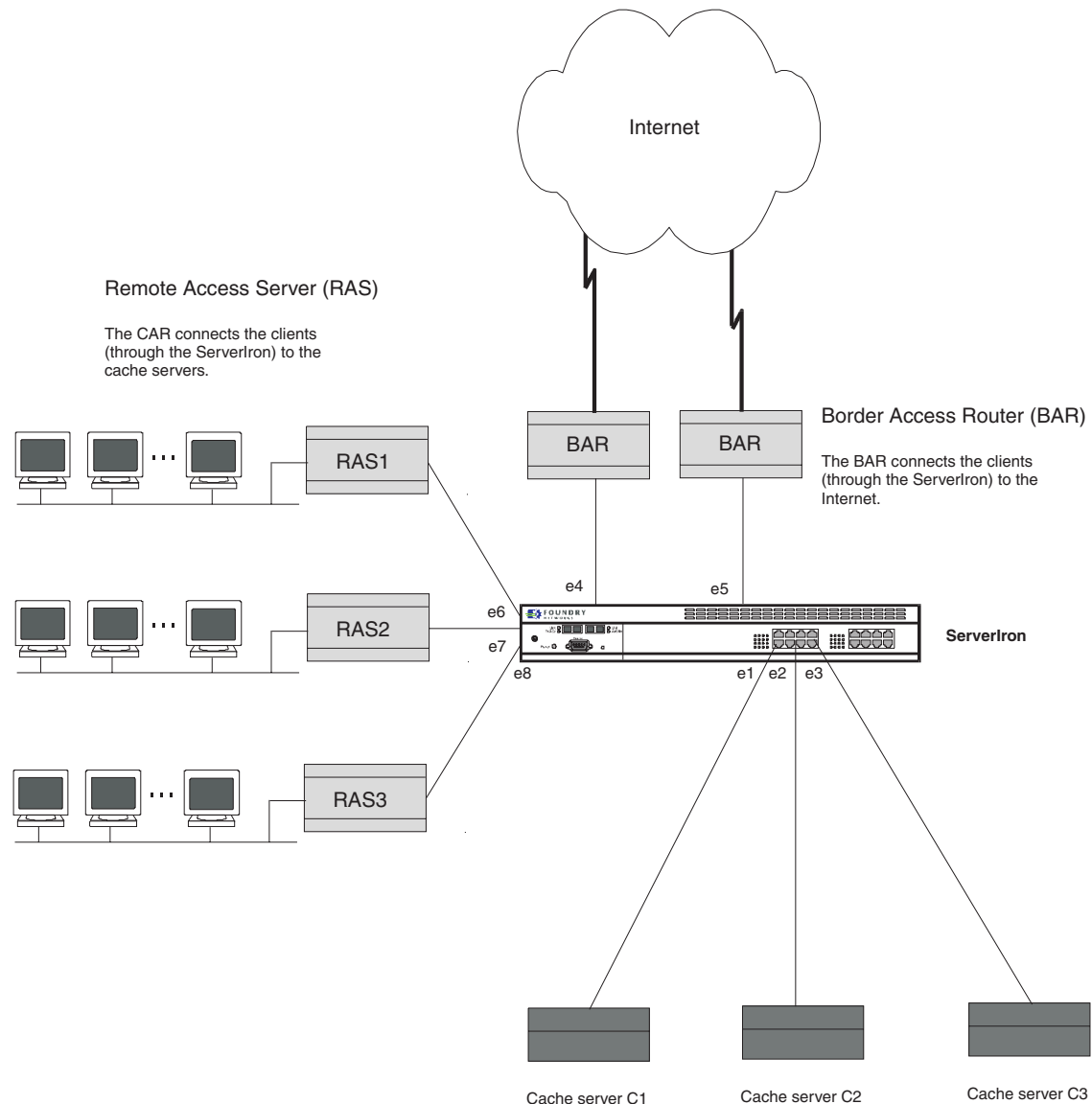
**Table 10.4: TCS Application Examples**

Application	See page...
Basic TCS configuration example	10-32
POP belonging to an ISP using caching to minimize WAN costs	10-35
Cache Route Optimization (CRO)	10-35
Policy-based cache switching	10-37
FastCache for asymmetric topologies	10-39
Policy-based Cache Failover (CFO)	10-41
Proxy Server Cache Load Balancing	10-42

### Basic TCS Configuration Example

Figure 10.13 shows a configuration in which HTTP traffic flows into a Point-of-Presence (POP) from remote access servers (RASs) and out of the POP to the Internet through a Border Area Router (BAR). The cache servers are labeled C1, C2, and C3.

Figure 10.13 Basic TCS configuration example



In the most basic setup, HTTP traffic flowing across the ServerIron, in any direction, is redirected to the cache servers. If a cache server has the requested content, the server returns the content to the client. If the cache server does not have the content, the cache server goes to the Internet to get the requested content, then caches the content and sends it to the client.

The client never accesses the Internet directly, unless all the cache servers in the cache group are unavailable. In that case, traffic flows across the ServerIron at Layer 2 and out to the Internet in the normal way.

In a transparent caching scheme, the ServerIron acts as the traffic redirector and the cache servers accept requests for any destination IP address. A cache server that accepts requests for any IP address are running in promiscuous mode. The client does not have to configure anything on their web browser. Thus, the caching is "transparent" to the client. It is this transparent characteristic that sets proxy-based caching and transparent caching apart.

In this example, suppose you want all traffic to be cached and you want to use the ServerIron's default settings. To configure the ServerIron for this example, you define the caches, assign them to cache groups, and apply an IP policy.

## IP Policies

For the simple case in which you want to cache everything no matter where it comes from or where it is going to, use a global policy.

```
ServerIron(config)# ip policy 1 cache tcp 80 global
```

By using a global policy, you can make rule 2 true for all ports. Rule 1 is true by default because all ports are in cache group 1. Any HTTP traffic flowing across the switch is redirected to the caches.

You can accomplish the same thing with a local policy. With local policies you have to first define and then apply the policy to the appropriate output ports. In this case, since you want to cache all traffic, you need to apply the policy to the RAS and BAR ports.

```
ServerIron(config)# ip policy 1 cache tcp 80 local
ServerIron(config)# int e 4
ServerIron(config-if-4)# ip-policy 1
ServerIron(config-if-4)# int e 5
ServerIron(config-if-5)# ip-policy 1
ServerIron(config-if-5)# int e 6
ServerIron(config-if-6)# ip-policy 1
ServerIron(config-if-6)# int e 7
ServerIron(config-if-7)# ip-policy 1
ServerIron(config-if-7)# int e 8
ServerIron(config-if-8)# ip-policy 1
```

---

**NOTE:** Note the subtle syntax difference between the commands to create a local policy and apply a policy to a port. If you leave the dash out of the command, the command does not work.

---

The local policies make rule 2 true for the BAR and RAS ports. Rule 1 is true by default. Local policies provide better control at the cost of more configuration steps. If you add a BAR to port 10, traffic destined for it is not redirected because you have not applied the policy to port 10. With a global policy, traffic is redirected automatically.

## Defining the Caches

To make caching work, you need to apply an IP policy and you need to define the caches and assign them to a cache-group. The example in the previous section shows how to apply the policy. You define cache servers as follows:

```
ServerIron(config)# server cache-name C1 11.11.11.11
ServerIron(config)# server cache-name C2 11.11.11.12
ServerIron(config)# server cache-name C3 11.11.11.13
```

The ServerIron ARPs for these addresses to determine which ports the caches are on.

## Defining the Cache Groups

A **cache group** is a collection of ServerIron input ports and cache servers. You can define up to four cache groups on a ServerIron. Each cache group can have a cache server farm with up to 256 caches. If a cache group has more than one cache server, the ServerIron distributes traffic among the servers using a hashing algorithm. (See “Defining Distribution of Web Requests Within a Cache Group” on page 10-12.)

All ports on the ServerIron are assigned to cache group 1 by default. Ports can be assigned to any cache group (only one at a time) or removed from all cache-groups. If a port is removed from all cache-groups, traffic entering on that port is not be redirected to a cache because rule 1 in this example is not true.

Once the caches have been defined, they must be associated (bound) with a particular cache group. The following CLI commands bind the cache servers shown in Figure 10.13 with a cache group.

```
ServerIron(config)# server cache-group 1
ServerIron(config-tc-1)# cache-name C1
ServerIron(config-tc-1)# cache-name C2
ServerIron(config-tc-1)# cache-name C3
```

## POP Belonging to an ISP Using Caching to Minimize WAN Costs

This example assumes a POP that belongs to an ISP. The RASs are actually remote access routers for customer dial-in. The ISP does not pay the phone company for access to the RASs; the ISP customers pay the phone company for this access. However, the ISP does pay for the WAN links connecting the BARs to the Internet. The ISP wants to introduce caching to improve user response time without the need to increase the size of the WAN links.

The ISP does not want to fill up the cache servers with content in its customer's web sites. The ISP wants to cache only the content on the other side of the BARs. The ISP wants only the traffic entering from a RAS destined for a BAR to be cached. The ISP does not want to cache RAS-to-RAS or BAR-to-RAS traffic.

In this example, the configuration requires more control than a global policy allows. Therefore, local policies are used. Only one cache group, the default cache group 1, is required.

To configure the ServerIron for this application, apply IP policies only to the BAR ports (4 and 5), define the caches, and place them in cache group 1. Here are the CLI commands for creating this configuration:

```
ServerIron(config)# server cache-name C1 11.11.11.11
ServerIron(config)# server cache-name C2 11.11.11.12
ServerIron(config)# server cache-name C3 11.11.11.13
ServerIron(config)# ip policy 1 cache tcp 80 local
ServerIron(config)# int e 4
ServerIron(config-if-4)# ip-policy 1
ServerIron(config-if-4)# int e 5
ServerIron(config-if-5)# ip-policy 1
ServerIron(config-if-5)# ser cache-group 1
ServerIron(config-tc-1)# cache-name c1
ServerIron(config-tc-1)# cache-name c2
ServerIron(config-tc-1)# cache-name c3
```

Traffic entering from a BAR destined for a RAS is not cached because rule 2 (output redirection enabled) is not true for the RAS ports. Traffic from RAS-to-RAS is not cached because rule 2 is false in this case as well. Traffic from RAS-to-BAR is cached because both rules are true.

Both rules are true for BAR-to-BAR traffic as well. This type of traffic rarely, if ever, occurs. However, if this type of traffic does occur and you do not want to cache the traffic, you cannot turn off the output policy on the BAR ports or nothing will get cached. Instead, make rule 1 false by removing the BAR ports from all cache groups. These ports are in the default cache group 1.

```
ServerIron(config)# int e 4
ServerIron(config-if-4)# no cache-group 1
ServerIron(config-if-4)# int e 5
ServerIron(config-if-5)# no cache-group 1
```

Now RAS-to-BAR traffic is still cached because the input ports are in the default cache group and the output ports have the IP policy applied. BAR-to-RAS and RAS-to-RAS traffic is not cached because rule 2 is still false. BAR-to-BAR traffic is not cached because rule 1 is false.

## Cache Route Optimization (CRO)

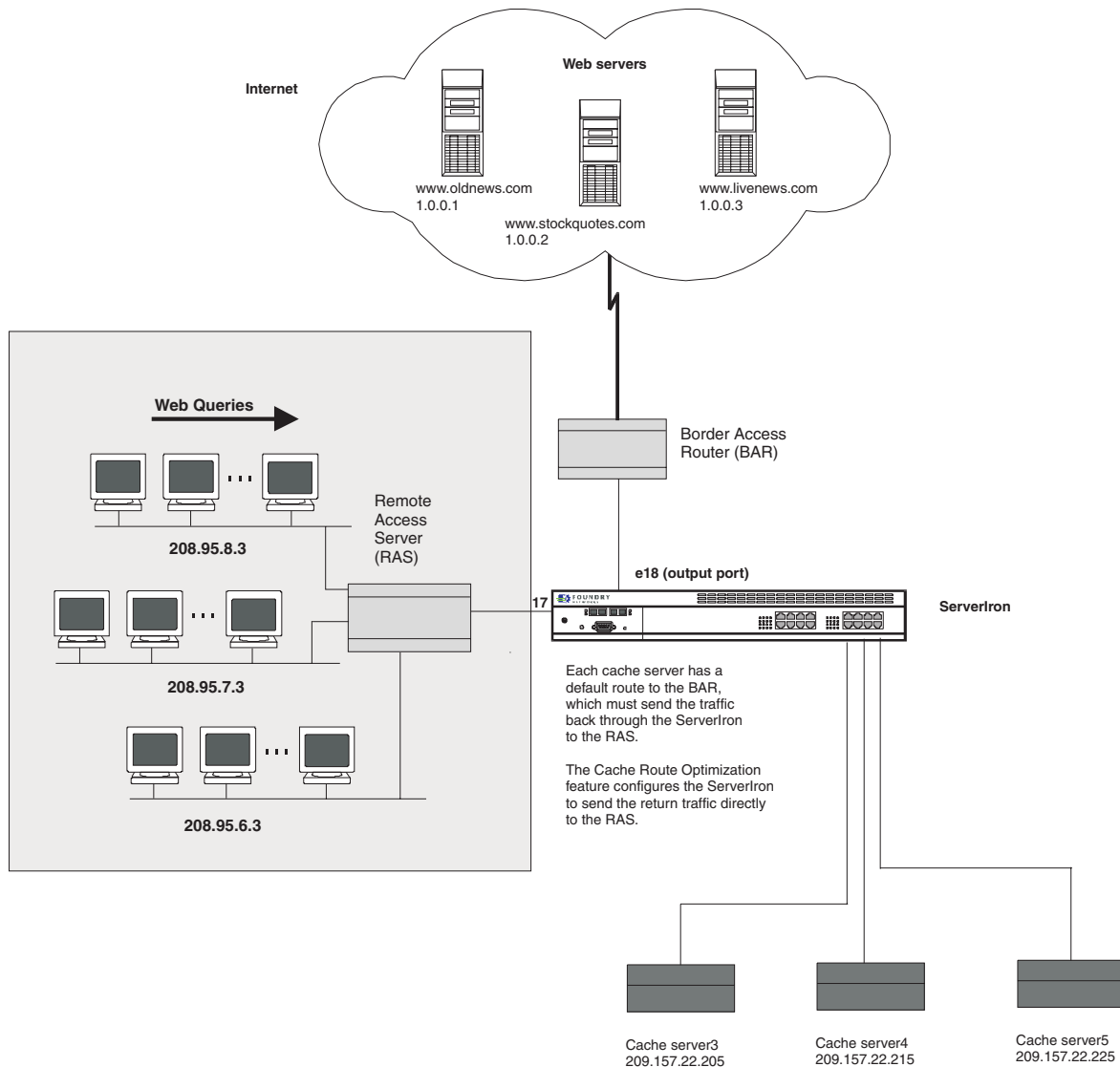
The Cache Route Optimization (CRO) feature solves a typical network topology dilemma, in which a cache server's default gateway is not the most direct route to the client. Figure 10.14 shows an example of a network with this topology.

In this example, return traffic from the cache servers passes through the ServerIron to the BAR because the BAR is the default gateway for the cache servers. However, the traffic is destined for the clients on the other side of the RAS. The ServerIron can switch the traffic at wire-speed, causing no perceivable response delays for the clients even if their return traffic must pass through the ServerIron twice. However, the client return traffic might add noticeable overhead to the BAR, especially if the BAR is also the default gateway for other devices on the network.

You can reduce the burden on the BAR by enabling CRO. This feature configures the ServerIron to use the information in its Layer 4 session table to recognize that the return traffic actually should go to the RAS instead of

the BAR, and send the return traffic directly to the RAS. Thus, the return traffic does not needlessly add overhead to the BAR.

**Figure 10.14 Cache route optimization**



To enable CRO for this configuration, enter the following command:

```
ServerIron(config)# server cache-router-offload
```

### Why ICMP Redirects Do Not Solve the Problem

The ServerIron redirects HTTP traffic destined for the Internet to the cache server. When the cache server responds to the client, it does so by sending its packets to its default gateway because the users are not in the same subnet as the cache server. However, at Layer 3, the packet is addressed to a client that is actually accessible through the RAS. The BAR knows the proper next hop router is the RAS, through a routing protocol, and retransmits the packet to the RAS, at Layer 2. The RAS forwards the packet to the client. Thus every packet to every client must go to the BAR and then be retransmitted. The BAR port is already carrying all the fetch and refresh traffic from that cache and this additional traffic can overload it.

The BAR does not send an ICMP redirect in this case, as you might expect. A router sends ICMP redirects only if the following conditions are met:

- The interface on which the packet comes into the router is the same as the interface on which the packet gets routed out.
- The route being used for the outgoing packet must not have been created or modified by an ICMP redirect, and must not be the router's default route.
- The sub-net of the source IP address is the same as the sub-net of the next-hop IP address of the routed packet.
- The packet must not be source routed.
- The router must be configured to send redirects.

The third rule is violated here because caches put the web server's address in the source address field rather than the cache's address. Thus in this scenario, the packet is retransmitted to the best next hop router (the RAS) but no ICMP redirect is sent.

### The ServerIron Solution

The ServerIron's CRO feature is a Layer 2 mechanism that solves the problem described above. When the cache server responds to a client, the first packet is forwarded to the BAR as discussed above. The BAR then retransmits the packet with the RAS as the destination MAC address and the BAR as the source MAC. The ServerIron examines the packet at Layer 4. The ServerIron finds a session table entry for this packet and knows it came from the cache server. The ServerIron knows the packet has been re-transmitted because the packet's source MAC address isn't the cache server's MAC address and the input port isn't the cache server's port. The ServerIron also recognizes that for this particular TCP session, it has seen the same packet with two different destination MAC addresses and knows that the second MAC address is the more appropriate one to use.

The ServerIron contains a state table that includes a field for a MAC address. Initially this field is blank. If the ServerIron sees that a packet has been re-transmitted, the ServerIron places the new destination MAC address (the RAS MAC address) in the state table. When subsequent packets are sent from the cache server, the ServerIron sees that there is a MAC address in the state table and replaces the destination MAC address with this address and forwards the packet.

### How Cache Route Optimization Works

Each TCP connection between the cache and a client is tracked by the ServerIron in a state table. The state table uses a key made up of the Layer 4 header: Source IP address, Source TCP port, Destination IP address, and Destination TCP port. The state table also has a field for a MAC address. This field is initially set to null (empty). When the cache server sends a packet a client, the ServerIron examines its Layer 4 header and checks to see whether it matches an entry in the state table. The ServerIron also examines the source MAC address to verify that the cache sent the packet. If the MAC address field in the state table is null, and it will be for the first packet, the ServerIron simply forwards the packet at Layer 2 to the cache's default gateway, the BAR.

When the packet is re-transmitted by the BAR, the ServerIron examines the Layer 4 header again, and sees that it matches an existing connection. The ServerIron also examines the source MAC address to be sure the cache server sent the packet. In this case, the source MAC address is the BAR's MAC, not the cache server's. The ServerIron concludes that this packet has been retransmitted and places the destination MAC address of the packet, the RAS's MAC, into the state table's MAC address field for this connection. Then the packet is forwarded to the RAS at Layer 2.

When the cache server transmits the next packet, the ServerIron compares its Layer 4 header to the state table and gets a match and now the entry has a MAC address in the MAC address field. The ServerIron replaces the destination address with the stored MAC address and transmits the packet at Layer 2 using the new "optimum" MAC address. Thus all packets except the first packet are sent directly to the optimum router.

Because this scheme works at the MAC layer, it is compatible with all routing protocols. Moreover, because the scheme is session specific, it can handle any number of RASs. When a session is terminated, the table entry is deleted and so is the "optimization". Thus changes in the network at Layer 3 are immediately implemented.

### Policy-Based Caching

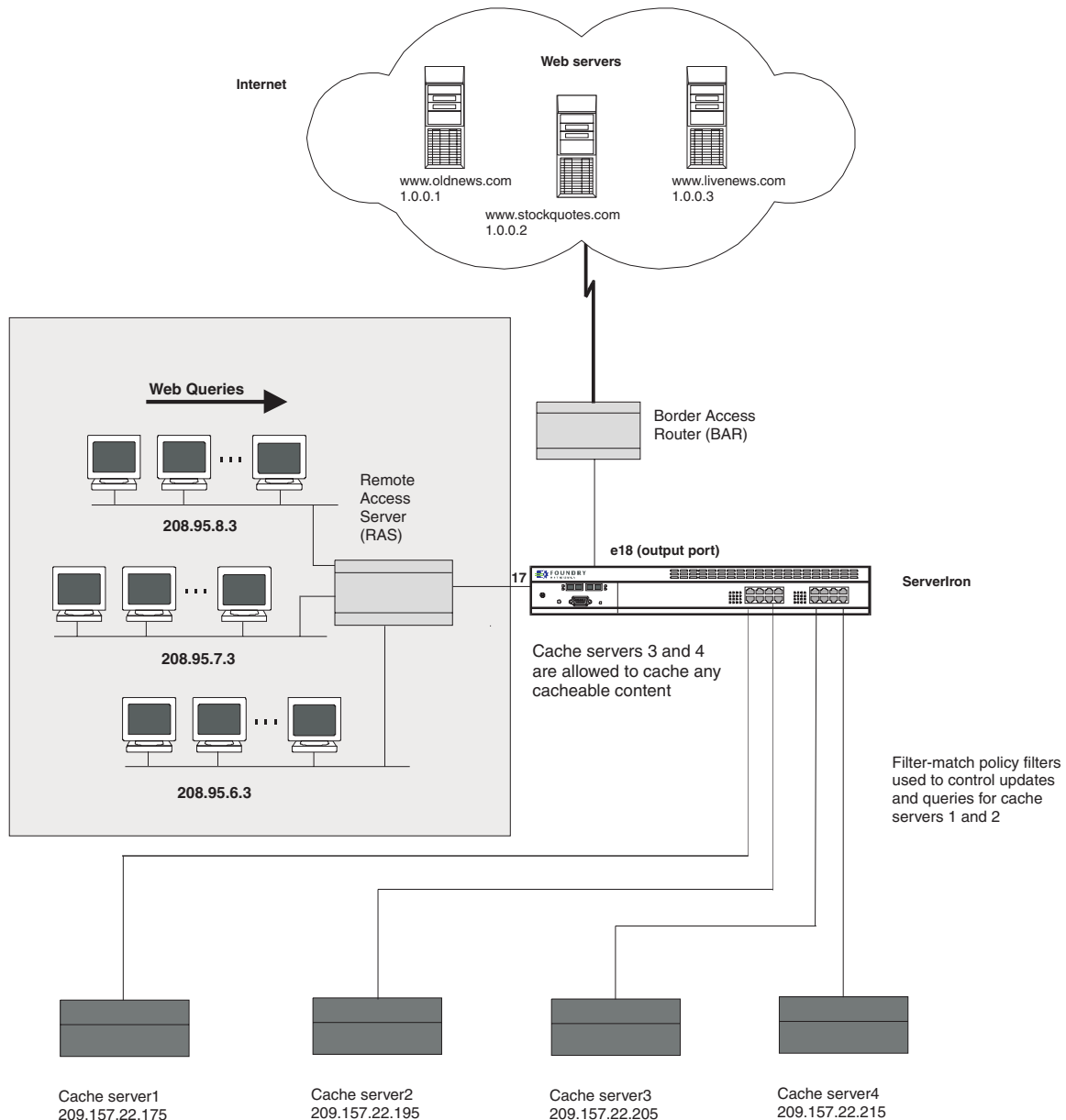
Normally, IP filters control whether the ServerIron redirects web requests from certain hosts or for certain sites to a cache server or sends the requests to the live sites on the Internet. **Policy-based caching** enables you to use IP filters to selectively cache some web sites but not others, *on specific cache servers*. For example, an ISP can

use a ServerIron configured for policy-based caching to redirect HTTP traffic to a series of web cache servers made by different vendors with different caching criteria.

Figure 10.15 shows a TCS configuration in which some of the cache servers are configured to cache any web sites that are cacheable, but other web cache servers are preconfigured with certain cached sites and should not cache other sites. In this application, the ISP wants to prevent new sites from being cached on the preconfigured cache servers, but still wants to allow new cache entries to be cached on the other cache servers.

In addition, the ISP wants to use filters to direct content that is received using the filters to the preconfigured cache servers. Not only that, the ISP wants to ensure smooth service by load balancing among the preconfigured cache servers.

**Figure 10.15 ServerIron configured to selectively cache contents based on cache server**



Here are the CLI commands for implementing the example configuration:

```
ServerIron(config)# server cache-name cacheserver1 209.157.22.175
```



```

ServerIron(config-rs-cacheserver1)# filter-match
ServerIron(config-rs-cacheserver1)# exit
ServerIron(config)# server cache-name cacheserver2 209.157.22.195
ServerIron(config-rs-cacheserver2)# filter-match
ServerIron(config-rs-cacheserver2)# exit
ServerIron(config)# server cache-name cacheserver3 209.157.22.205
ServerIron(config-rs-cacheserver3)# exit
ServerIron(config)# server cache-name cacheserver4 209.157.22.215
ServerIron(config-rs-cacheserver4)# exit
ServerIron(config)# ip filter 1 permit any 1.0.0.1 255.255.255.255 tcp eq 80
ServerIron(config)# ip filter 1023 deny any 1.0.0.3 255.255.255.255 tcp eq 80
ServerIron(config)# ip filter 1024 permit any any

```

These commands configure the cache servers shown in Figure 10.15, then create filters.

Filter 1 allows content from the www.oldnews.com web server to be cached only on cache servers 1 and 2, which are preconfigured with this web site. The content is not cached on the other cache servers.

Filter 1023 prevents any of the cache servers from receiving content from www.livenews.com. The ServerIron always passes traffic for this site directly to the site itself on the Internet.

Filter 1024 ensures that cache servers that are allowed to receive content from any cacheable server receive that content unless the server is explicitly blocked by another filter. Cache servers 3 and 4 do not have the **filter-match** parameter and are allowed to receive content from any cacheable server, except the site filtered out by filter 1023 and the site directed only to cache servers 1 and 2 by filter 1.

---

**NOTE:** Traffic that matches filter 1 for cache servers 1 and 2 is load balanced using the least-connections or weighted least-connections metric, which otherwise apply only to Server Load Balancing. See “Load Balancing Method (Predictor)” on page 6-24. Traffic for the other cache servers is partitioned according to the hash feature. (See “Distribution Algorithm” on page 10-12.)

---

## FastCache

Traffic in typical TCS configurations passes through the ServerIron both from the client to the cache and from the cache to the client. The ServerIron uses the cache responses to the client to diagnose the health of the cache server. If the cache server responds to the client requests the ServerIron redirects to the cache server, the ServerIron knows that the cache server is healthy. However, if the cache server stops sending replies to the client requests, the ServerIron assumes that the cache server is down and stops redirecting requests to that cache server.

Some configurations are asymmetric—traffic from the cache server to the client does not pass back through the ServerIron. For example, caches that support multiple NICs might at the same time support only one default gateway. Figure 10.16 shows a configuration in which a cache server’s default gateway is configured to go to the customer access router (RAS) instead of the ServerIron. In this configuration, the ServerIron does not see cache responses to client requests. Because the ServerIron does not see responses coming from the cache server, the ServerIron assumes that the cache server is down and stops redirecting requests to that cache server.

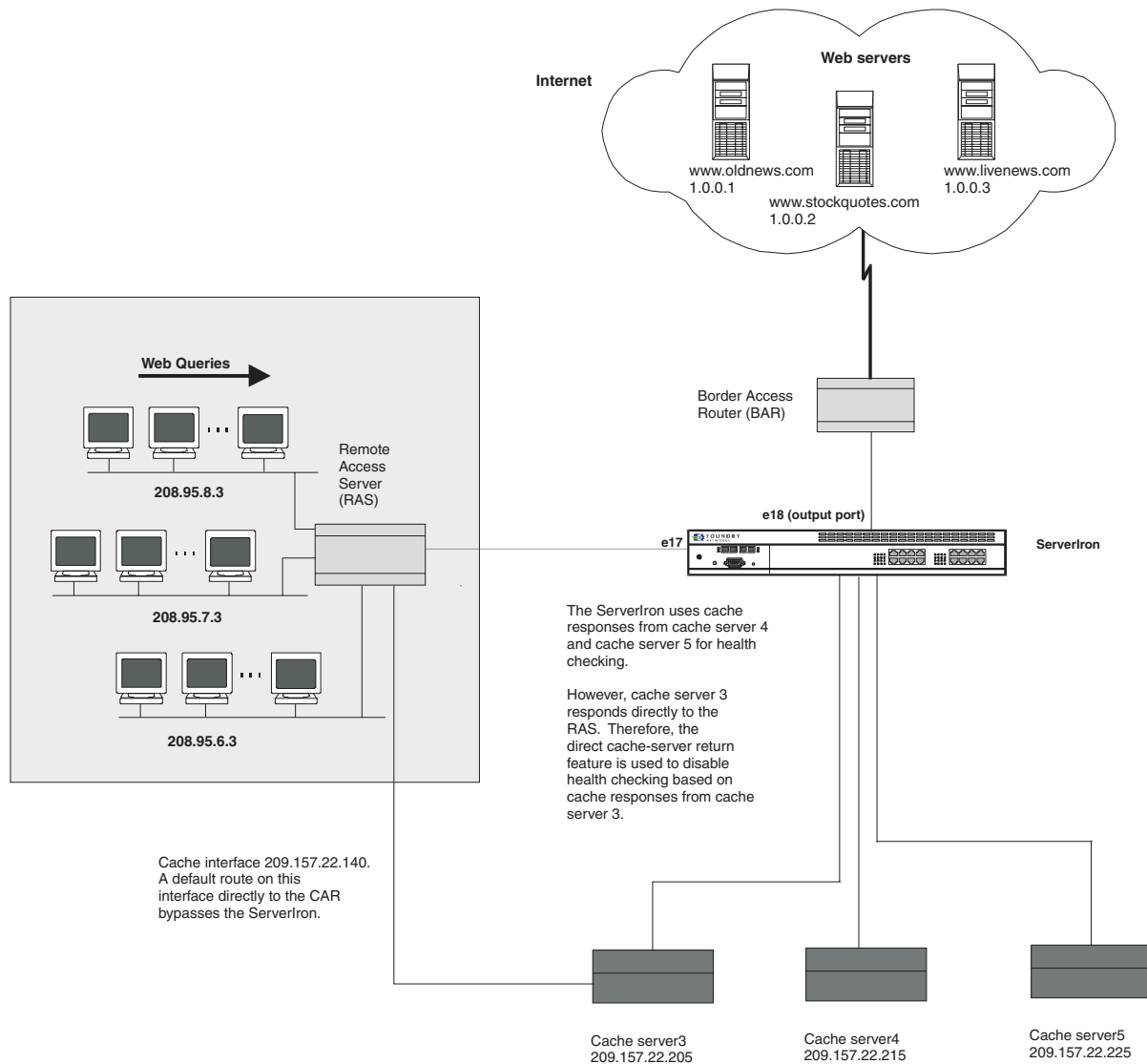
You can override this behavior by enabling the FastCache feature. This feature configures the ServerIron to continue redirecting client requests to a cache server even though the ServerIron does not see responses from the cache server. You enable the feature individually for real servers.

---

**NOTE:** Even when use the FastCache feature, the ServerIron still performs a Layer 3 health check by regularly pinging the cache server. In addition, you can continue to use HTTP health checking. See “Layer 7 Health Checks” on page 12-3.

---

Figure 10.16 FastCache feature used for asymmetric topology



Here are the commands for configuring the ServerIron for the topology shown in Figure 10.16. The line that enables the FastCache feature is shown in bold.

```
ServerIron(config)# server cache-name cacheserver3 209.157.22.205
ServerIron(config-rs-cacheserver3)# asymmetric
ServerIron(config-rs-cacheserver3)# exit
ServerIron(config)# server cache-name cacheserver4 209.157.22.215
ServerIron(config-rs-cacheserver4)# exit
ServerIron(config)# server cache-name cacheserver5 209.157.22.225
ServerIron(config-rs-cacheserver5)# exit
```

This example assumes that the cache contains the contents requested by the client. However, if the cache does not contain the requested page, the cache tries to get the page from the live web site. In this case, the source address for the request is the IP address of the cache server, instead of the IP address of the client. Moreover, this behavior can result in a loop from the cache server to the RAS to the ServerIron and back to the cache server.

To prevent this situation from occurring:

- Define the other interface on the cache server as a cache, but do not place the cache in a cache group.
- Define an IP filter to prevent the cache associated with the source IP address of the cache server from being cached. The filter ensures that the ServerIron directs all traffic addressed to the cache server's other interface (the one with the default route to the RAS) directly to the Internet, instead of back to the cache.

## Policy-Based Cache Failover (CFO)

In some TCS configurations, the ServerIron is connected to the clients and also to the Internet through the same router. Moreover, in some cases the router contains a policy to forward HTTP requests to a next-hop IP address (virtual IP address) if the packet containing the request matches a filter configured in the router. CFO prevents client requests from becoming lost in a "black hole" when the cache servers are unavailable. When you configure the ServerIron for CFO, the ServerIron forwards client requests back to the router for forwarding to the Internet. Thus, clients still receive the requested content even though the cache servers are unavailable.

Normally, cache groups on the ServerIron do not have virtual IP addresses. Instead, the ServerIron selects a cache server from the cache group that contains the port to which the router is connected. Within the cache group, the ServerIron uses a hashing algorithm to select a specific cache server.

---

**NOTE:** The virtual servers in SLB use virtual IP addresses, but TCS does not use virtual IP addresses unless you are using CFO.

---

To configure CFO, make sure you do the following:

1. Set up the router and aim the policy on the router at the virtual address on the ServerIron rather than at the address of the cache.
2. Define the cache or caches on the ServerIron and place them into cache group 1.
3. Define the virtual IP address in cache group 1.
4. Define the IP cache policy as a global cache.

---

**NOTE:** For CFO, you must define a global policy, not a local policy.

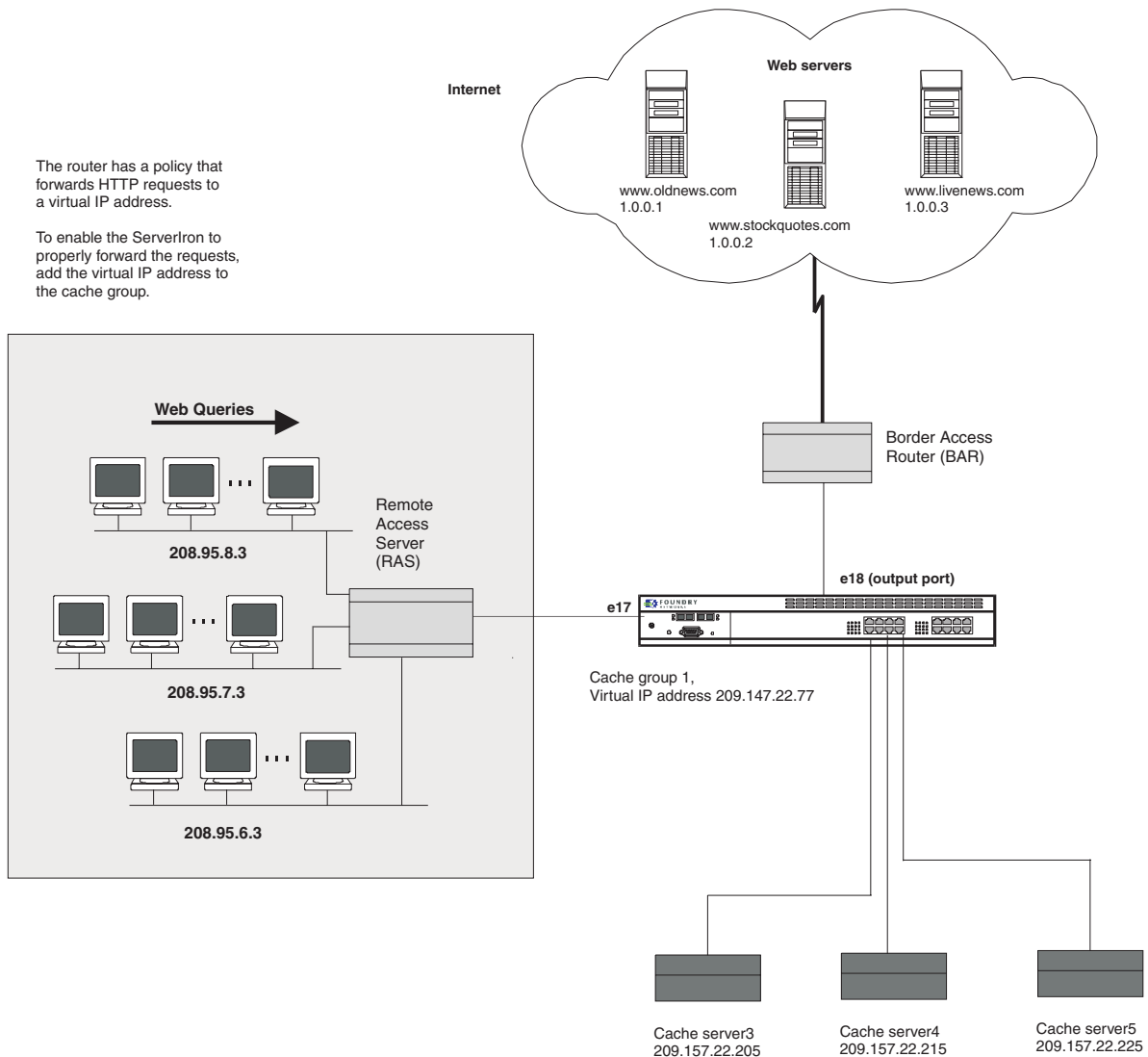
---

When you add the virtual IP address to the cache group:

- If the cache server to which the ServerIron sends the HTTP traffic has the requested page, the cache server sends the page back to the client, typically through the ServerIron. (This is the normal behavior regardless of whether you have added a virtual IP address.)
- If the cache server is unavailable or does not have the page and thus attempts to send the request back through the ServerIron to the Internet, the ServerIron sends the request to the router for forwarding to the Internet. If the virtual IP address is not configured on the ServerIron, the ServerIron drops the request from the cache server.

Figure 10.17 shows an example of a configuration that requires CFO.

**Figure 10.17 Configuration using policy-based Cache Failover (CFO)**



Here are the CLI commands for adding a virtual IP address to a cache group. Add the virtual IP address to which your router forwards the clients' HTTP requests.

```
ServerIron(config)# ip policy 1 cache tcp 80 global
ServerIron(config)# server cache-group 1
ServerIron(config-tc-1)# virtual-ip 209.157.22.77
```

## Proxy Server Cache Load Balancing

Proxy Server Cache Load Balancing relieves clients who have configured their web browsers to point to a proxy server from the need to reconfigure their browsers. When you configure the ServerIron for this feature, the ServerIron performs TCS for clients whose browsers do use a proxy and for clients whose browsers do not use a proxy:

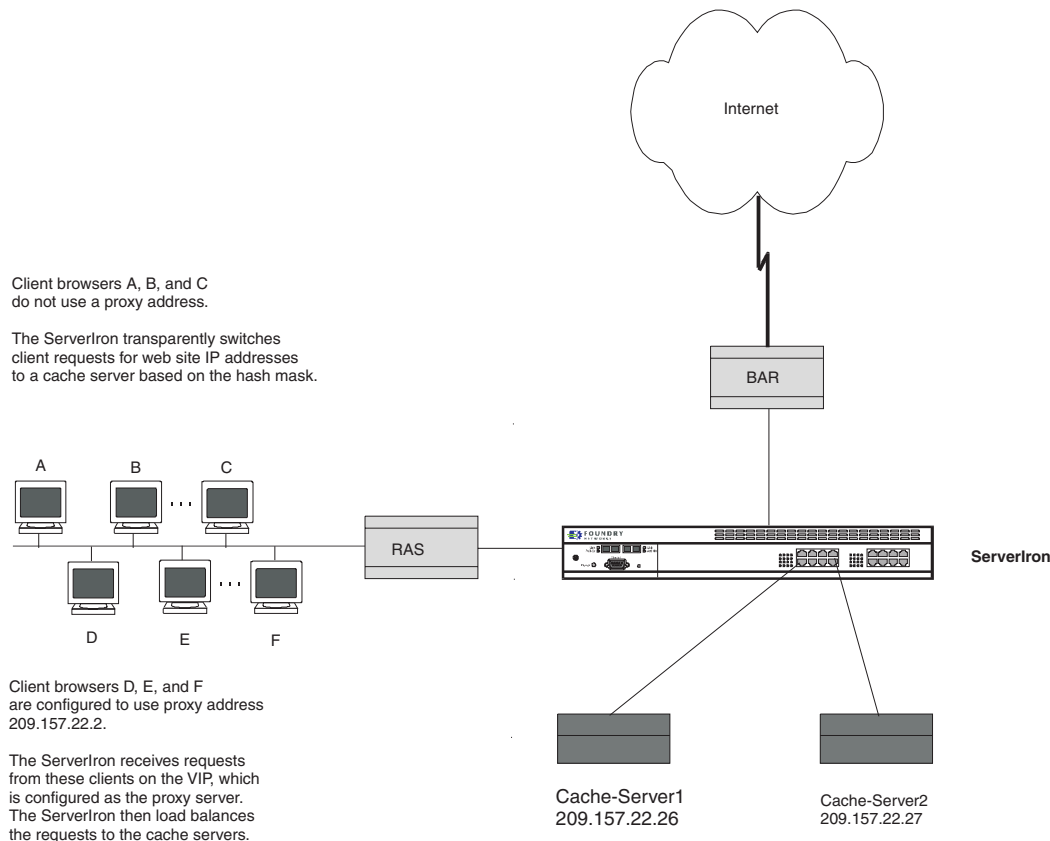
- For clients whose browsers do not use a proxy, the ServerIron performs transparent TCS, using the normal hash mechanism to map requests to a cache server based on the source and destination information in the mask and the IP address of the requested site.
- For clients whose browsers use a proxy, the ServerIron load balances the requests across the cache servers. Although this is different from the hash mechanism used for transparent TCS, the results for the client are

exactly the same. The ServerIron sends the request to a cache server that either has the requested content and sends it back to the client or does not have the requested content but quickly obtains it from the Internet, then sends it back to the client. In addition, the hash mechanism not only distributes traffic, but also ensures that duplication of content is minimized. The hash mechanism minimizes duplication by ensuring that a particular web site is always cached on the same cache server.

In either case, the ServerIron provides the requested content to the client.

Figure 10.18 shows an example of a TCS configuration in which some clients have browsers configured to use a proxy while other clients' browsers are not thus configured.

**Figure 10.18 Example Proxy Server Cache Load Balancing Configuration**



As shown in Figure 10.18, some clients' web browsers are configured to use proxy IP address 209.157.22.2, while other client's web browsers are not configured to use a proxy server. You can configure the ServerIron to satisfy both sets of clients.

To configure Proxy Server Cache Load Balancing:

- Add the cache servers as customary, using the **server cache-name** <string> <ip-addr> command.
- Add the HTTP ports and configure port-specific health check parameters at the Cache Server level, using the **port http** | <num> commands.
- Create the proxy virtual IP address (VIP) and bind the HTTP ports of the cache servers to the VIP. Use the **server virtual-name** <string> <ip-addr> and **bind...** commands.
- Add the cache servers to a cache group using the **server cache-group 1** command.
- Save the configuration changes to the startup-config file using the **write memory** command.

**NOTE:** If you have already configured your cache servers and cache group, you do not need to change their configuration. You only need to add the VIP for the proxy and bind the HTTP ports to it, then save the changes.

---

To configure the ServerIron for the example shown in Figure 10.18 on page 10-43, enter the following commands on the ServerIron:

```
ServerIron(config)# server port 4199
ServerIron(config-port-4199)# tcp
ServerIron(config-port-4199)# exit
ServerIron(config)# server port 8080
ServerIron(config-port-8080)# tcp
ServerIron(config-port-8080)# exit
```

The commands above add port profiles for the two HTTP ports in this example that are using port numbers other than the well-known port 80: 4199 and 8080. The **tcp** command at each port's configuration level is required. If you do not identify the ports as TCP ports, the ServerIron assumes the ports are UDP ports and thus does not use an appropriate health check for the ports. You do not need to add a port profile for port 80, since that is the well-known HTTP port.

```
ServerIron(config)# server cache-name Cache-Server1 209.157.22.26
ServerIron(config-Cache-Server1)# port 4199
ServerIron(config-Cache-Server1)# port 8080
ServerIron(config-Cache-Server1)# port http
ServerIron(config-Cache-Server1)# exit
ServerIron(config)# server cache-name Cache-Server2 209.157.22.27
ServerIron(config-Cache-Server2)# port 4199
ServerIron(config-Cache-Server2)# port 8080
ServerIron(config-Cache-Server2)# port http
ServerIron(config-Cache-Server2)# exit
```

The commands above add cache servers Cache-Server1 and Cache-Server2. The **port** commands add the HTTP ports to the cache servers. This example does not include optional modification of the HTTP health check parameters for specific servers. For information about customizing an HTTP health check for a specific server, see "Configuring Health Checks" on page 12-1.

```
ServerIron(config)# server virtual-name Proxy 209.157.22.2
ServerIron(config-vs-Proxy)# port 4199 sticky
ServerIron(config-vs-Proxy)# port 8080 sticky
ServerIron(config-vs-Proxy)# bind 4199 Cache-Server1 4199 Cache-Server2 4199
ServerIron(config-vs-Proxy)# bind 8080 Cache-Server1 8080 Cache-Server2 8080
ServerIron(config-vs-Proxy)# exit
```

The commands above configure a virtual IP address (VIP) to take the place of the Proxy IP address to which some of the client browsers are directing their web requests. The IP address specified with the **server virtual-name** command is the IP address that is configured as the proxy on some clients' web browsers. The **port 4199 sticky** and **port 8080 sticky** commands add the ports and also make them "sticky". When a port is sticky, once a client session is established on the port, the ServerIron's load balancing mechanism (used for the proxy) sends subsequent packets in the same session to the same cache server. The **sticky** parameter is not required in this configuration but it can streamline cache performance by keeping client sessions on the same cache servers.

The **bind** commands create table entries in the ServerIron that associate the cache servers and their HTTP ports with the Proxy VIP.

```
ServerIron(config)# server cache-group 1
ServerIron(config-tc-1)# cache-name Cache-Server1
ServerIron(config-tc-1)# cache-name Cache-Server2
ServerIron(config-tc-1)# write mem
```

The commands above add the cache servers to a cache group, then save the configuration changes to the ServerIron's startup-config file.

## Content Aware Cache Switching

Content aware cache switching (URL switching in a TCS environment) uses information in the header of an HTTP request to determine how or if content should be retrieved from a cache server. Using the text in a URL string, the ServerIron sends a request from a client to a cache server or to the Internet according to user-defined policies.

You can configure content aware cache switching on the ServerIron to do the following:

- Group cache servers by content; for example, GIF files can be cached on one cache server and HTML files on another
- Cause HTTP requests containing a given URL string always to go to the same cache server, minimizing content duplication among cache servers
- Use information in the URL string or Host header field of an HTTP request to determine how the requested content should be cached
- Explicitly direct requests for dynamic content to the Internet, rather than to a cache server
- Use directives in the HTTP 1.0 or 1.1 header to determine whether requested content should be cached

The following sections discuss how URL switching operates in a TCS environment and present some sample configurations that demonstrate the features of content aware cache switching.

### How URL Switching Works

**URL switching** is the ServerIron's ability to direct HTTP requests to a server, or group of servers, using information in the text of a URL string. The ServerIron examines the contents of a URL string and makes a decision about where to send the packet based on selection criteria in user-defined policies. If text in the URL string matches the selection criteria, the HTTP request is sent to a server group specified in the policy.

---

**NOTE:** "URL string" is defined as *the contents of the Request-URI part of the Request-Line in an HTTP request message*. This information usually consists of the absolute pathname (directory and filename) of a resource. For example:

/doc/ServerIron/1199/url\_switching.html

The URL string can also be the input to a process running on a remote server. For example:

/quote.cgi?s=FDRY&d=1d

The network location of the resource is specified in the Host header field in an HTTP request message. For example:

Host: www.foundrynet.com

The ServerIron can examine both the URL string and Host header field when determining where to send the HTTP request. See RFC 1945 or RFC 2616 for more information on HTTP request messages.

---

The selection criteria in a policy can be a string of characters starting from the beginning of the URL string, end of the URL string, or within any part of the URL string. For example, selection criteria can be a URL string that starts with the text "/home". In a TCS environment, when a client sends an HTTP request that has a URL string beginning with the text "/home", the policy can direct that request to a specific group of cache servers (or to another URL switching policy for additional matching).

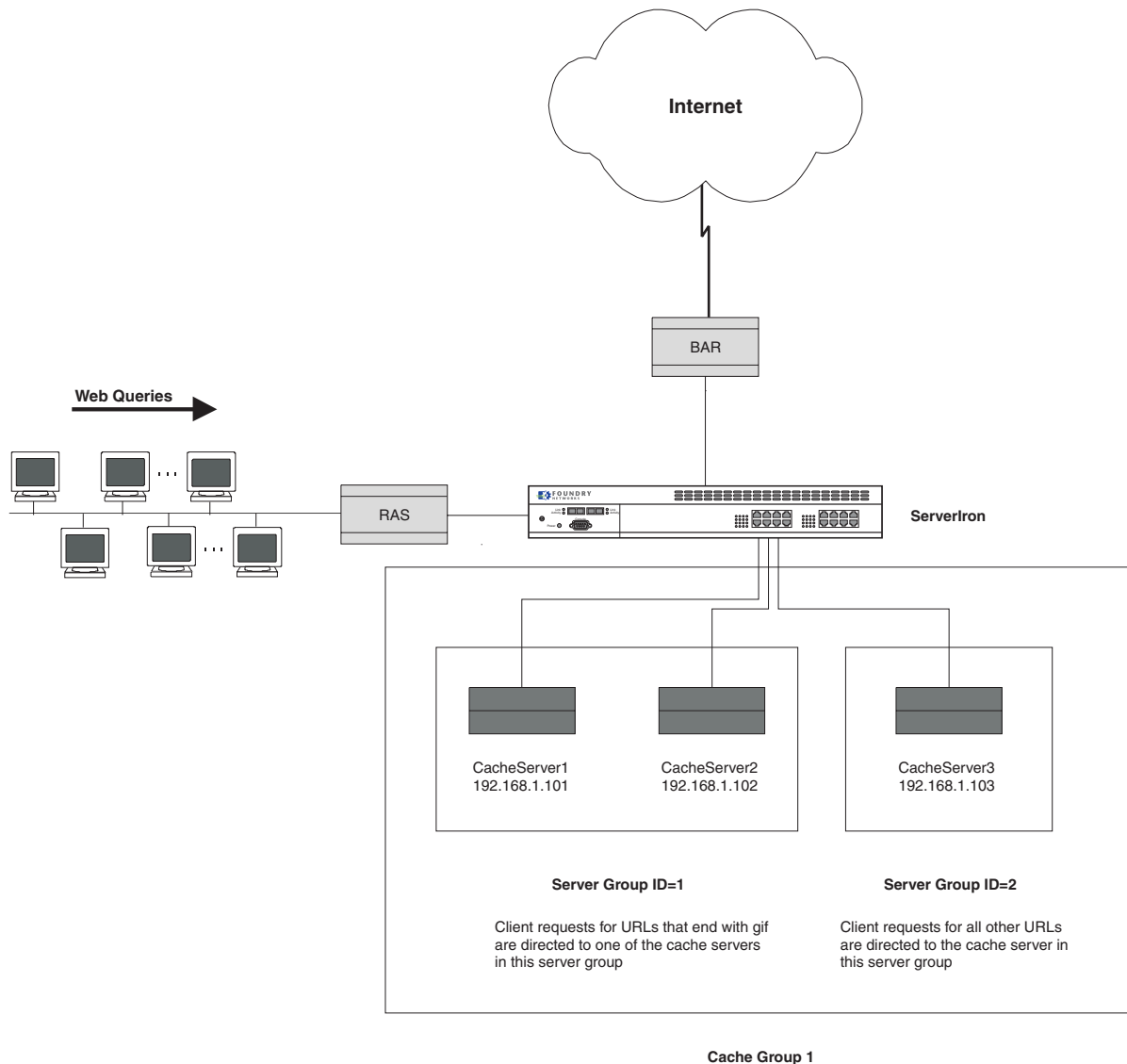
## Basic Example of Content Aware Cache Switching

The diagram in Figure 10.19 illustrates a configuration that uses content aware cache switching to cache GIF files on one set of cache servers and different kinds of files on another set.

In this configuration, cache group 1 consists of three cache servers. CacheServer1 and CacheServer2 are allocated to server group ID = 1, and CacheServer3 is allocated to server group ID = 2. The ServerIron has URL switching policies in place that cause HTTP requests to be directed to the cache servers as follows:

- HTTP requests containing URL strings that end with the text ".gif" are sent to one of the cache servers in server group ID = 1.
- If a URL string does not end with the text ".gif", the HTTP request is sent to the cache server in server group ID = 2.

**Figure 10.19 Content aware cache switching**



The first time a client requests a URL that ends with ".gif" (for example, /home/main/banner.gif) the following events take place:



1. Since the URL ends with "gif", a URL switching policy on the ServerIron directs the request to one of the cache servers in Server Group ID=1.
2. When a server group consists of more than one cache server, the ServerIron uses a hashing algorithm (see "URL String Hashing" on page 11-35 for details) to select one of the cache servers, and directs the request to the selected cache server.
3. Since this is the first time the content is requested, the selected cache server doesn't have the content stored, so the cache server retrieves it from the Internet.
4. The cache server receives the content, caches it, and sends it to the requesting client.

The next time a client requests the content, the following events take place:

1. Since the URL begins with "gif", the URL switching policy directs the request to one of the cache servers in Server Group ID=1.
2. The ServerIron hashes the URL string, selecting the same server it selected previously.
3. This time the cache server has the content and doesn't have to go to the Internet to get it; it sends the cached content to the requesting client.

Setting up content aware cache switching consists of the following steps:

1. Enabling TCS on the ServerIron
2. Setting up URL switching policies
3. Configuring the cache servers
4. Assigning the cache servers to a cache group

These tasks are described in the following sections.

### **Enabling TCS**

To enable TCS on all interfaces (globally) of the ServerIron shown in Figure 10.19, enter the following command:

```
ServerIron(config)# ip policy 1 cache tcp 80 global
```

**Syntax:** ip policy <index> cache | normal | high tcp | udp <tcp/udp-portnum> global | local

### **Setting up the URL Switching Policies**

URL switching policies define selection criteria for URL strings and specify what happens when a URL string matches the selection criteria. In content aware cache switching, if an HTTP request contains a URL string that matches a policy's selection criteria, the HTTP request can be sent to a load-balanced cache server group or to another policy for additional matching.

---

**NOTE:** The URL switching policies discussed in this section apply to the example in Figure 10.19 on page 10-46.

---

### **USING THE CLI**

The following commands define a URL switching policy called p1.

```
ServerIron(config)# url-map p1
ServerIron(config-url-p1)# method suffix
ServerIron(config-url-p1)# match "gif" 1
ServerIron(config-url-p1)# default p2
ServerIron(config-url-p1)# exit
```

**Syntax:** url-map <policy-name>

**Syntax:** method prefix | suffix | pattern

**Syntax:** match "<selection-criteria>" <server-group-id> | <policy-name>

**Syntax:** default <server-group-id> | <policy-name>

The **url-map p1** command sets the name of the policy and enters the URL switching CLI level.

The **method suffix** command specifies what kind of matching the policy does on the selection criteria. Three kinds of matching methods are supported:

- prefix**      Compares the selection criteria to the beginning of the URL string.
- suffix**      Compares the selection criteria to the end of the URL string.
- pattern**      Looks for the selection criteria anywhere within the URL string.

The **match "gif" 1** command consists of two parts. The first part specifies the selection criteria, which can be up to 80 characters in length; the second part indicates what to do when the URL string matches the selection criteria – send the HTTP request to a server group, send the request to the Internet, or match the URL string against another policy. In this example, the selection criteria is the text string "gif". Since the matching method is **suffix**, the policy looks at the end of the URL string. If the URL string ends with the text "gif", then the URL string meets the selection criteria.

---

**NOTE:** In addition to using text as selection criteria, you can use an asterisk (\*) as a wildcard character to specify one or more characters at the end of a URL string. For example, using "/ho\*" as the selection criteria matches /home, /hotels, and /home/main/index.html.

If you are using the suffix matching method, you cannot use an asterisk (\*) as a wildcard character. The asterisk wildcard character is valid for the prefix and pattern matching methods only.

---

If the URL string meets the selection criteria, the second part of the **match** command specifies what to do with the HTTP request. In this example, the **1** in the command causes the HTTP request to be sent to the cache server group whose ID = 1. Specifying **0** in the **match** command causes the request to be directed to the Internet. A URL switching policy can contain multiple **match** commands, each with different selection criteria.

---

**NOTE:** You can also specify a URL switching policy name instead of a server group ID. In this case, if part of the URL string matches the selection criteria, the remaining text of the URL string (that is, the text that was not matched by the selection criteria) is evaluated by the specified policy.

---

The **default p2** command specifies what happens when the URL string does not meet any of the selection criteria in a URL switching policy's **match** command. As with a **match** command, you can specify either a server group ID number or another URL switching policy. In this example, if a URL string does not match the selection criteria in policy p1, it is sent to policy p2 for evaluation.

The following commands define URL switching policy p2 for the example in Figure 10.19.

```
ServerIron(config)# url-map p2
ServerIron(config-url-p2)# default 2
ServerIron(config-url-p2)# exit
```

The single command in policy p2, **default 2**, causes all HTTP requests encountered by the policy to be sent to server group ID = 2. In this configuration, policy p2 will encounter requests passed to it by policy p1; that is, all the requests that do not have "gif" at the end of the URL string.

---

**NOTE:** As the diagram in Figure 10.19 illustrates, there is only one cache server in server group ID = 2. Even so, the **match** command must refer to a server group, rather than an actual cache server. Server groups can consist of one or more cache servers.

---

### **Configuring the Cache Servers**

The cache servers return the content to the requesting clients. When configuring content aware cache switching, you place the cache servers into logical server groups. URL switching policies direct HTTP requests to one of the cache servers in these logical groups.

A server group can contain one or more cache servers. When a server group consists of more than one cache server, the ServerIron uses a hashing algorithm to select one of the cache servers, and directs the request to the selected cache server. (See "URL String Hashing" on page 11-35 for details on the hashing algorithm.) When configuring content aware cache switching, you establish the IP address of each cache server and specify the server group to which it belongs.

### USING THE CLI

To configure CacheServer1 in Figure 10.19 on page 10-46:

```
ServerIron(config)# server cache-name CacheServer1 192.168.1.101
ServerIron(config-rs-CacheServer1)# port http group-id 1 1
ServerIron(config-rs-CacheServer1)# exit
```

**Syntax:** port http group-id <server-group-id-pairs>

The **port http group-id** command indicates the server group(s) to which the cache server belongs. The server group is expressed as a pair of numbers, indicating a range of server group IDs. The first number is the lowest-numbered server group ID, and the second is the highest-numbered server group ID. For example, if a cache server belongs only to the server group with ID = 1, the last two numbers in the **port http group-id** command would be **1 1**. (Note the space between the two numbers.) If a cache server belongs to server groups 1 – 10, the last two numbers in the command would be **1 10**. Valid numbers for server group IDs are 0 – 1023.

To include a cache server in groups that are not consecutively numbered, you can enter up to four server group ID pairs. For example, to include a cache server in groups 1 – 5 and 11 – 15, you would enter the following command:

```
ServerIron(config-rs-CacheServer1)# port http group-id 1 5 11 15
```

You can also specify the server group ID pairs on separate lines; for example:

```
ServerIron(config-rs-CacheServer1)# port http group-id 1 5
ServerIron(config-rs-CacheServer1)# port http group-id 11 15
```

The configuration for the remaining cache servers in Figure 10.19 is shown below. These commands place CacheServer2 in server group ID = 1 (along with CacheServer1) and CacheServer3 in server group ID = 2.

```
ServerIron(config)# server cache-name CacheServer2 192.168.1.102
ServerIron(config-rs-CacheServer2)# port http group-id 1 1
ServerIron(config-rs-CacheServer2)# exit

ServerIron(config)# server cache-name CacheServer3 192.168.1.103
ServerIron(config-rs-CacheServer3)# port http group-id 2 2
ServerIron(config-rs-CacheServer3)# exit
```

### Assigning the Cache Servers to a Cache Group

To activate content aware cache switching (as in Figure 10.19), you create a cache group, assign the cache servers to that group, and specify a URL switching policy to be active for the cache group. For example:

```
ServerIron(config)# server cache-group 1
ServerIron(config-tc-1)# cache-name CacheServer1
ServerIron(config-tc-1)# cache-name CacheServer2
ServerIron(config-tc-1)# cache-name CacheServer3
ServerIron(config-tc-1)# url-map p1
ServerIron(config-tc-1)# url-switch
```

**Syntax:** url-map <policy-name>

**Syntax:** url-switch

The **url-map** command specifies a URL switching policy to be active for this cache group. If you configure more than one URL switching policy, the policies must be linked together. In this example, policy p1 may send text to policy p2. Thus, the two policies are linked together. Up to 100 URL switching policies can be linked in this way.

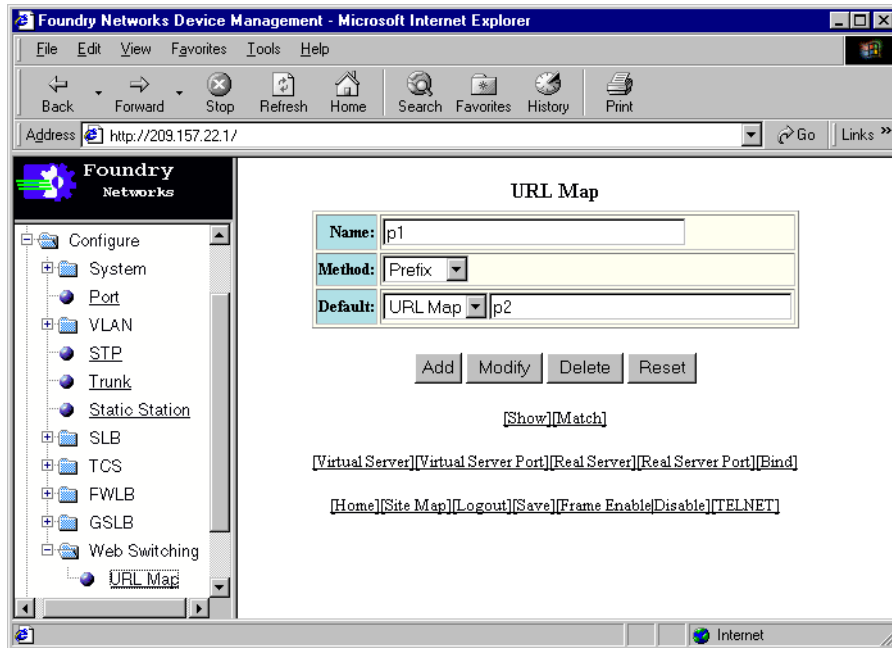
The **url-switch** command activates URL switching for this cache group. You must have already defined the URL switching policies before entering this command.

### USING THE WEB MANAGEMENT INTERFACE

To implement content aware cache switching, use the following procedure.

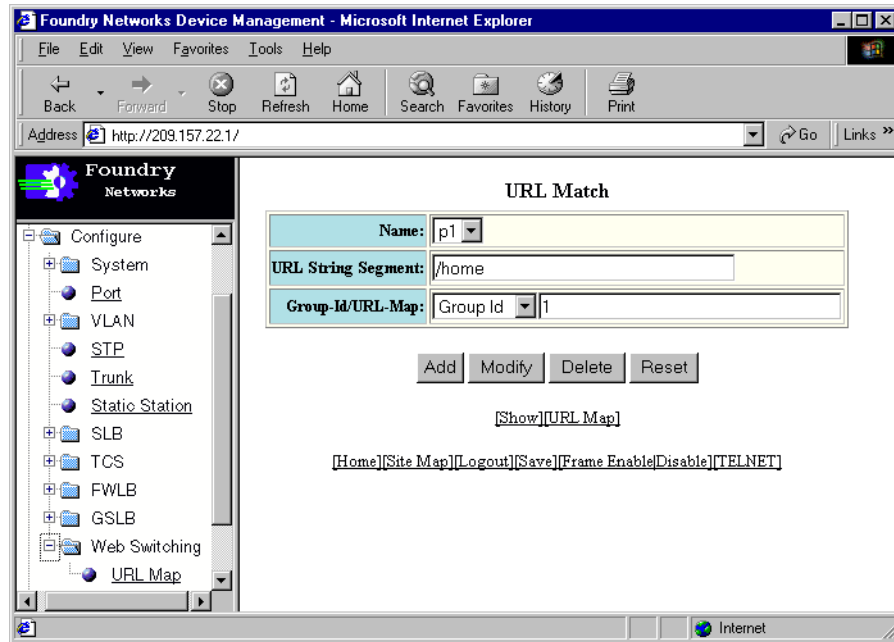
1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.

3. Click on the plus sign next to Web Switching in the tree view to expand the list of Layer 7 switching option links.
4. Select the URL Map link. The following panel is displayed:



5. In the Name field, enter the name of this URL switching policy; for example, p1
6. Select the matching method for this URL switching policy from the Method pulldown menu. Specify one of the following:
  - Prefix** Compares the selection criteria to the beginning of the URL string.
  - Suffix** Compares the selection criteria to the end of the URL string.
  - Pattern** Looks for the selection criteria anywhere within the URL string.
7. In the Default field, specify what happens when the URL string does not meet any of the policy's selection criteria. The HTTP request can be directed to a cache server group, or the URL string can be matched against another policy.
  - To direct the HTTP request to a cache server group, select Group Id from the pulldown menu and enter the ID of the cache server group.
  - To match the URL string against another URL switching policy, select URL Map from the pulldown menu and enter the name of the policy.
8. Click the Add button to add the URL switching policy to the device's running-config file.

9. Click the Match link. The following panel is displayed:



10. Select the URL switching policy from the Name pulldown menu.
11. In the URL String Segment field, enter the selection criteria for the policy. The selection criteria can be up to 80 characters in length.

**NOTE:** In addition to using text as selection criteria, you can use an asterisk (\*) as a wildcard character to specify one or more characters at the end of a URL string. For example, using "/ho\*" as the selection criteria matches /home, /hotels, and /home/main/index.html.

12. In the Group-Id/URL-Map field, specify what happens when the URL string meets the policy's selection criteria. The HTTP request can be directed to a cache server group, or the URL string can be matched against another policy.
- To direct the HTTP request to a cache server group, select Group Id from the pulldown menu and enter the ID of the cache server group.
  - To match the URL string against another URL switching policy, select URL Map from the pulldown menu and enter the name of the policy.
13. Click Add to save the changes to the device's running-config file.
14. Repeat Steps 11 – 13 for each set of selection criteria in the policy.
15. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
16. Click on the plus sign next to TCS in the tree view to expand the list of Transparent Cache Switching option links.
17. Select the Web Cache link from the menu. A panel such as the one shown below will appear.

**NOTE:** If cache servers are already configured on the ServerIron, a summary panel listing the configured cache servers is listed instead. In this case, select the [Add Web Cache](#) link.

Foundry Networks Device Management - Microsoft Internet Explorer

Address: http://209.157.22.1/

**Foundry Networks**

Configure

- System
- Port
- VLAN
- STP
- Trunk
- Static Station
- SLB
- TCS
  - Cache Router Offload
  - Select Members
  - Server Group
  - Web Cache
  - Web Cache Port
- Command

**Web Cache**

Server Name:	frownland
Server IP:	207.95.5.11
Maximum Connections:	1000000
Weight:	1
Asymmetric:	<input type="checkbox"/>
Source NAT:	<input type="checkbox"/>
Destination NAT:	<input type="checkbox"/>
Filter Match:	<input type="checkbox"/>

Add Modify Delete Reset

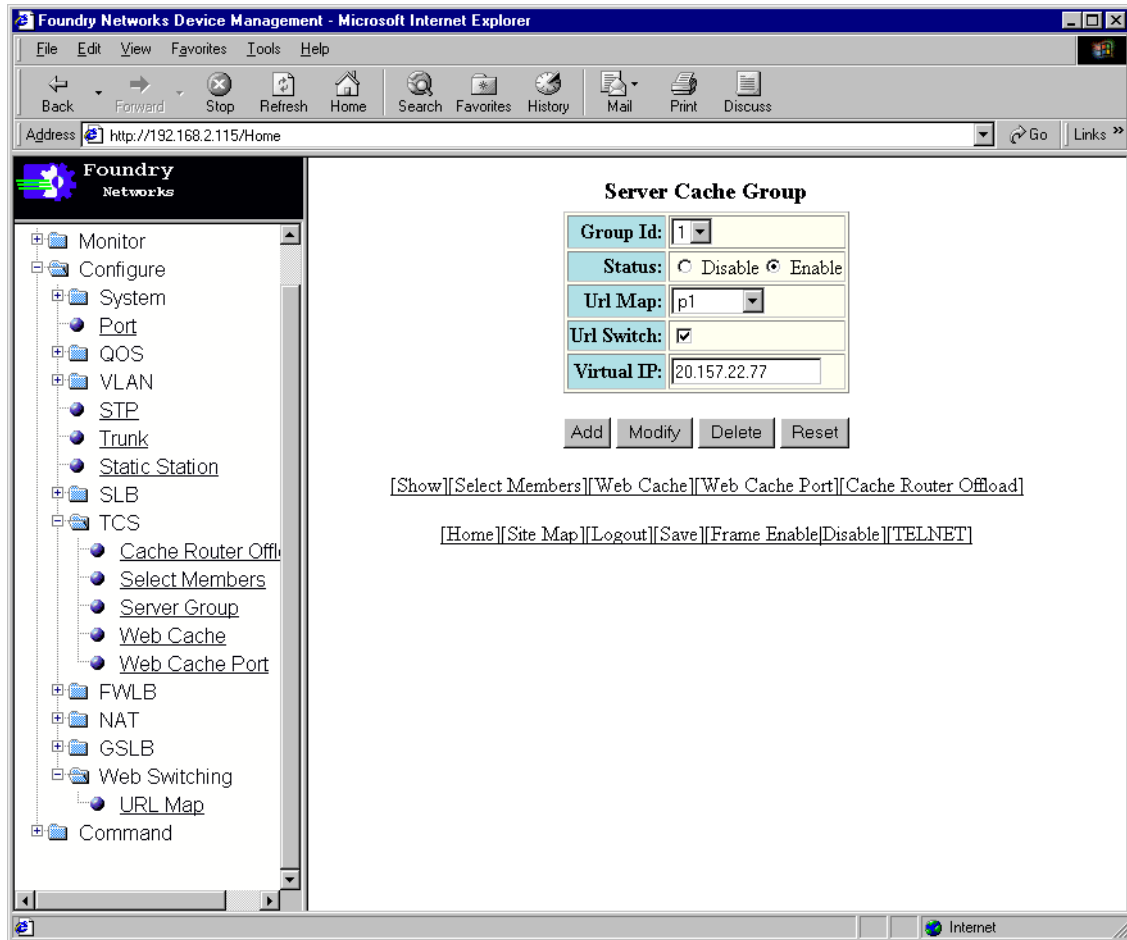
[Show]

[Server Group][Cache Router Offload][Web Cache Port][Layer 4 QOS]

[Home][Site Map][Logout][Save][Frame Enable][Disable][TELNET]

18. Enter the name of the cache server in the Server Name field.
19. Enter the IP address in the Server IP field.

20. Select the Server Group link at the bottom of the panel. The following panel will appear.



21. Select a group ID from the pulldown menu. Possible values are 1 – 4.
22. Select a URL switching policy from the URL Map pulldown menu.
23. Click the checkbox next to URL Switch.
24. Click the Add button to save the changes to the device's running-config file.

**NOTE:** When you click Add, the ServerIron examines the configured URL switching policies to see that each policy called by another policy actually exists. If a policy is called but does not exist, the ServerIron displays a message indicating the URL map is not valid. If you see this message, make sure the URL switching policies are configured correctly.

25. Select the Web Cache Port link at the bottom of the panel. The following panel will appear.

The screenshot shows a web browser window titled "Foundry Networks Device Management - Microsoft Internet Explorer". The address bar shows "http://192.168.2.115/Home". The left sidebar contains a tree view with the following items: Monitor, Configure, System, Port, QOS, VLAN, STP, Trunk, Static Station, SLB, TCS, Cache Router Offl, Select Members, Server Group, Web Cache, Web Cache Port (selected), FWLB, NAT, GSLB, Web Switching, URL Map, and Command. The main content area is titled "Web Cache Port" and contains the following configuration fields:

- Name:** CacheServer1
- TCP/UDP Port:** HTTP (dropdown menu) with a "User Define" button.
- Status:** ☐ Disable ☒ Enable
- Keep Alive:** ☐
- HTTP Parameters:**
  - \*Method:** HEAD (dropdown menu)
  - \*URL:** (text input field)
  - \*Status Code:** (text input field)
- DNS Parameters:**
  - +DNS Zone:** (text input field)
  - +Addr Query:** (text input field)
- Group Id Range:**

From	To
1	5
11	15

At the bottom of the panel are four buttons: Add, Modify, Delete, and Reset.

26. Select the cache server from the Name pulldown menu.

27. Select HTTP from the TCP/UDP Port pulldown menu.

28. In the Group Id Range fields, enter the server group(s) to which the cache server belongs. You specify the server group IDs in terms of one or more ranges:

- In the From field, enter the lowest-numbered server group ID to which this cache server belongs.
- In the To field, enter the highest-numbered server group ID to which this cache server belongs.

You can enter up to four ranges of server group IDs. Valid numbers for server group IDs are 0 – 1023. If the cache server belongs to only one server group, you can enter the server group ID in the first From field and leave the other Group Id Range fields blank.

29. Click the Add button to save the changes to the device's running-config file.

30. Select the Save link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Directing Requests Using the HTTP Host Header Field

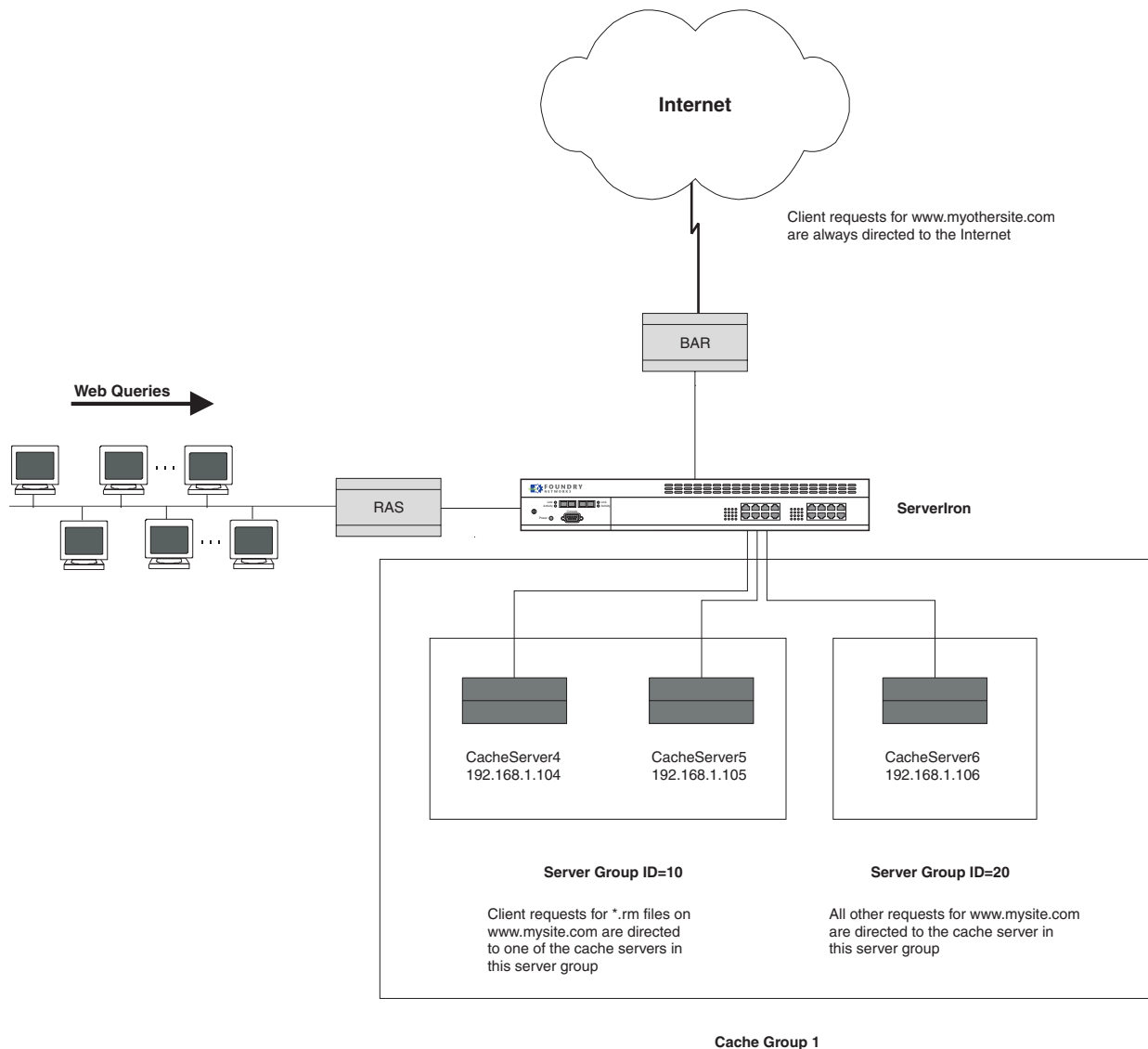
Figure 10.20 on page 10-55 illustrates another example of content aware cache switching. This example demonstrates how the ServerIron can examine the Host header field in addition to the URL string when determining where to send an HTTP request.



This configuration specifies how requests for two web sites, [www.mysite.com](http://www.mysite.com) and [www.myothersite.com](http://www.myothersite.com), are directed to cache servers:

- HTTP requests for [www.mysite.com](http://www.mysite.com) go to a cache server in one of two server groups, depending on the content of the URL string. Requests for [www.mysite.com](http://www.mysite.com) that have URL strings ending with the text ".rm" are sent to server group ID = 10. All other HTTP requests for [www.mysite.com](http://www.mysite.com) are sent to server group ID = 20.
- All requests for [www.myothersite.com](http://www.myothersite.com) are directed to the Internet, not to a cache server.
- Requests for neither [www.mysite.com](http://www.mysite.com) nor [www.myothersite.com](http://www.myothersite.com) are sent to a cache server in server group ID = 20.

**Figure 10.20 URL switching example using the HTTP host header field**



The following sections explain how to set up this configuration.

### Enabling TCS

To enable TCS on all interfaces (globally) of the ServerIron shown in Figure 10.20, enter the following command:

```
ServerIron(config)# ip policy 1 cache tcp 80 global
```

**Syntax:** ip policy <index> cache | normal | high tcp | udp <tcp/udp-portnum> global | local

### Setting up the URL Switching Policies

To implement the configuration in Figure 10.20, you would create three URL switching policies:

- PolicyA and policyB apply to HTTP requests for www.mysite.com
- PolicyB applies to requests for www.mysite.com that do not have URL strings ending with "rm", as well as requests for neither www.mysite.com nor www.myothersite.com
- PolicyZ applies to HTTP requests for www.myothersite.com

### USING THE CLI

The following commands define policyA:

```
ServerIron(config)# url-map policyA
ServerIron(config-url-policyA)# method suffix
ServerIron(config-url-policyA)# match "rm" 10
ServerIron(config-url-policyA)# default policyB
ServerIron(config-url-policyA)# exit
```

The **method suffix** command causes the policy to examine the last part of the URL string.

The **match "rm" 10** command looks for URL strings that end with the text "rm"; for example, /home/content/show.rm. These HTTP requests are sent to server group ID = 10.

The **default policyB** command sends HTTP requests that do not meet the selection criteria in policyA's **match** command to policyB for evaluation.

The following commands define policyB:

```
ServerIron(config)# url-map policyB
ServerIron(config-url-policyB)# default 20
ServerIron(config-url-policyB)# exit
```

The single command in policyB, **default 20**, causes all HTTP requests encountered by the policy to be sent to server group ID = 20. In this configuration, policyB will encounter requests passed to it by policyA (that is, all the requests for www.mysite.com that do not have "rm" at the end of the URL string) as well as requests not for www.mysite.com or www.myothersite.com.

The following commands define policyZ:

```
ServerIron(config)# url-map policyZ
ServerIron(config-url-policyZ)# default 0
ServerIron(config-url-policyZ)# exit
```

This policy simply directs all HTTP requests it encounters to the Internet. The only requests that policyZ will encounter are those for www.myothersite.com.

### Configuring the Cache Servers

To place CacheServer4 and CacheServer5 in Figure 10.20 on page 10-55 into server group ID = 10, enter the following commands:

```
ServerIron(config)# server cache-name CacheServer4 192.168.1.104
ServerIron(config-rs-CacheServer4)# port http group-id 10 10
ServerIron(config-rs-CacheServer4)# exit

ServerIron(config)# server cache-name CacheServer5 192.168.1.105
ServerIron(config-rs-CacheServer5)# port http group-id 10 10
ServerIron(config-rs-CacheServer5)# exit
```

To place CacheServer6 into server group ID = 20, enter the following commands:

```
ServerIron(config)# server cache-name CacheServer6 192.168.1.106
ServerIron(config-rs-CacheServer6)# port http group-id 20 20
ServerIron(config-rs-CacheServer6)# exit
```

### Assigning the Cache Servers to a Cache Group

To activate the configuration shown in Figure 10.20, enter the following commands:

```
ServerIron(config)# server cache-group 1
ServerIron(config-tc-1)# cache-name CacheServer4
ServerIron(config-tc-1)# cache-name CacheServer5
ServerIron(config-tc-1)# cache-name CacheServer6
ServerIron(config-tc-1)# url-host-id www.mysite.com policyA
ServerIron(config-tc-1)# url-host-id www.myothersite.com policyZ
ServerIron(config-tc-1)# url-map policyB
ServerIron(config-tc-1)# url-switch
```

**Syntax:** url-host-id <host> <policy-name>

The **url-host-id www.mysite.com policyA** command causes HTTP requests for www.mysite.com to be evaluated by policyA.

The **url-host-id www.myothersite.com policyZ** command causes HTTP requests for www.myothersite.com to be evaluated by policyZ.

If a request is for neither www.mysite.com nor www.myothersite.com, then the request is evaluated by policyB. In this example, the **url-map policyB** command functions similarly to the **default** command in a URL switching policy, sending requests that don't meet the other selection criteria to a "catch-all" policy.

### Using a Wildcard Character in the url-host-id Command

You can use an asterisk (\*) as a wildcard character to specify one or more characters at the beginning of the Host header field. For example, specifying "\*.com" as the <host> in the **url-host-id** command matches all requests for hosts ending with .com. The following commands illustrate the use of the wildcard character in the **url-host-id** command.

```
ServerIron(config)# server cache-group 1
ServerIron(config-tc-1)# cache-name CacheServer4
ServerIron(config-tc-1)# cache-name CacheServer5
ServerIron(config-tc-1)# cache-name CacheServer6
ServerIron(config-tc-1)# url-host-id *.com policyA
ServerIron(config-tc-1)# url-host-id www.myothersite.com policyZ
ServerIron(config-tc-1)# url-map policyB
ServerIron(config-tc-1)# url-switch
```

In this configuration, the **url-host-id \*.com policyA** command causes HTTP requests for any site ending in .com to be evaluated by policyA. Note that when there are multiple **url-host-id** commands in a cache group's configuration, the ServerIron favors an exact match over a wildcard match. In the sample configuration above, any requests for www.myothersite.com are evaluated by policyZ, not policyA.

### Configuring Policies for Dynamic Content

For dynamic Web pages, such as Active Server Pages, it may be preferable not to cache the content. You can configure URL switching policies on the ServerIron that cause requests for these kinds of pages to bypass the cache servers and go directly to the Internet.

In addition, the ServerIron examines directives in the HTTP 1.0 or 1.1 header to determine whether a request should be sent to the cache servers or to the Internet. When this feature is enabled (the default), a request is sent to the origin server regardless of the URL string if one of the following is true:

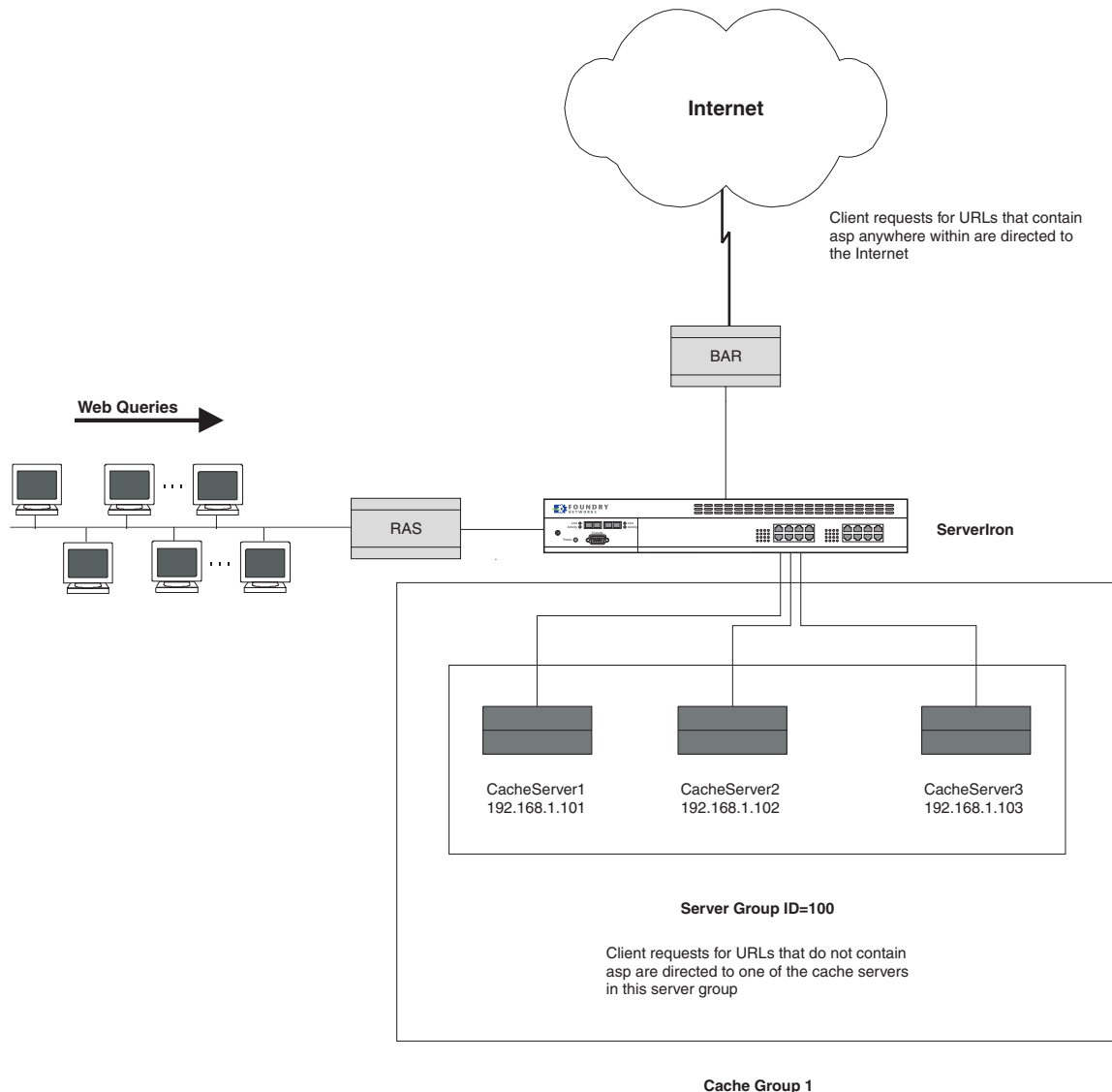
- The request contains a pragma:no-cache header (HTTP 1.0 requests)
- The Cache-Control header in the request contains a no-cache directive (HTTP 1.1 requests)

In the configuration in Figure 10.21 on page 10-58, the ServerIron has URL switching policies in place that cause HTTP requests to be directed to the cache servers as follows:

- Requests that have URL strings with the text "asp" anywhere within go directly to the Internet
- Requests for all other content are directed to one of the cache servers in server group ID = 100
- HTTP 1.0 requests that have a pragma:no-cache header are sent to the Internet regardless of the URL string

- HTTP 1.1 requests that have a Cache-Control header containing a no-cache directive are sent to the Internet regardless of the URL string

**Figure 10.21 Sending requests for Active Server Pages to the Internet**



The following sections explain how to set up this configuration.

#### **Enabling TCS**

To enable TCS on all interfaces (globally) of the ServerIron shown in Figure 10.21, enter the following command:

```
ServerIron(config)# ip policy 1 cache tcp 80 global
```

**Syntax:** ip policy <index> cache | normal | high tcp | udp <tcp/udp-portnum> global | local

#### **Setting up the URL Switching Policies**

To implement the configuration in Figure 10.21, you would create a URL switching policy that sends all requests containing URL strings ending with "asp" directly to the Internet, bypassing the cache servers. All other requests are sent to one of the cache servers in server group ID = 100.

### USING THE CLI

The following commands define a URL switching policy called policyA1:

```

ServerIron(config)# url-map policyA1
ServerIron(config-url-policyA1)# method pattern
ServerIron(config-url-policyA1)# match "asp" 0
ServerIron(config-url-policyA1)# default 100
ServerIron(config-url-policyA1)# exit

```

The **method pattern** command causes the policy to look for the selection criteria anywhere within the URL string.

The **match "asp" 0** command looks for URL strings that contain the text "asp"; for example, /active/q.asp?In=fdry. These HTTP requests are sent to the Internet.

The **default 100** command sends HTTP requests that do not meet the selection criteria in policyA1's **match** command server group ID = 100.

### Configuring the Cache Servers

To place the cache servers in Figure 10.21 on page 10-58 into server group ID = 100, enter the following commands:

```

ServerIron(config)# server cache-name CacheServer1 192.168.1.101
ServerIron(config-rs-CacheServer1)# port http group-id 100 100
ServerIron(config-rs-CacheServer1)# exit

ServerIron(config)# server cache-name CacheServer5 192.168.1.102
ServerIron(config-rs-CacheServer5)# port http group-id 100 100
ServerIron(config-rs-CacheServer5)# exit

ServerIron(config)# server cache-name CacheServer6 192.168.1.103
ServerIron(config-rs-CacheServer6)# port http group-id 100 100
ServerIron(config-rs-CacheServer6)# exit

```

### Assigning the Cache Servers to a Cache Group

To activate the configuration shown in Figure 10.21, enter the following commands:

```

ServerIron(config)# server cache-group 1
ServerIron(config-tc-1)# cache-name CacheServer1
ServerIron(config-tc-1)# cache-name CacheServer2
ServerIron(config-tc-1)# cache-name CacheServer3
ServerIron(config-tc-1)# http-cache-control
ServerIron(config-tc-1)# url-map policyA1
ServerIron(config-tc-1)# url-switch

```

**Syntax:** [no] http-cache-control

The **http-cache-control** command ensures that HTTP 1.0 requests that have a pragma:no-cache header and HTTP 1.1 requests that have a Cache-Control header containing a no-cache directive are sent to the Internet. This is the default behavior. To configure the ServerIron to ignore the pragma:no-cache or Cache-Control header in an HTTP request, use the **no http-cache-control** command.

### Using HTTP 1.1 for Connections to Cache Servers

When the ServerIron receives an HTTP request from a client, it uses its content aware cache switching configuration to determine to which cache server it should send the request. The ServerIron then establishes a TCP connection with the selected cache server and sends it the request.

If the request sent from the client to the ServerIron uses HTTP version 1.1, the ServerIron downgrades the HTTP version to 1.0 when it sends the request to the cache server. If you want to use HTTP 1.1 for the connection between the ServerIron and the cache servers, you can prevent the ServerIron from downgrading the HTTP version to 1.0 by entering the following commands:

```

ServerIron(config)# server cache-group 1
ServerIron(config-vs-tc-1)# no-http-downgrade
ServerIron(config-vs-tc-1)# exit

```

**Syntax:** no-http-downgrade

In HTTP version 1.0 (RFC 1945), a client can send only one request per TCP connection; in HTTP version 1.1 (RFC 2616) a client can send multiple requests per TCP connection. If the ServerIron sends requests to a cache server using HTTP 1.1, all the requests in the TCP connection are sent to the same cache server that the ServerIron selected using the first request, regardless of the contents of the URL string in the subsequent requests.

### Dropping Requests when a Server Group is Unavailable

By default, if none of the cache servers in a server group are available, the requests are directed to one of the other server groups configured on the device. You can change this default behavior so that requests are dropped rather than directed to another server group. To do this, enter the following commands:

```
ServerIron(config)# server cache-group 1
ServerIron(config-tc-1)# no-group-failover
ServerIron(config-tc-1)# exit
```

**Syntax:** no-group-failover

### Streaming Media Support

TCS can be used with streaming media content. The RTSP, MMS, and Real streaming media protocols are supported. The source NAT and destination NAT features are applied correctly to streams using these protocols, both for the parent TCP connection as well as the actual data stream.

To configure TCS for streaming media content, specify a streaming media port (RTSP, PNM, or MMS) as part of the definition of the cache server and configure an **ip policy** statement for the specified port.

For example, to configure TCS for the RTSP protocol, enter commands such as the following:

```
ServerIron(config)# server cache-name CacheServer1 192.168.1.101
ServerIron(config-rs-CacheServer1)# port rtsp
ServerIron(config-rs-CacheServer1)# exit
ServerIron(config)# ip policy 1 cache tcp rtsp global
```

If you use TCS with MMS (TCP port 1755 with random UDP ports) or PNM (TCP port 7070), you must specify port 0 in the **ip policy** command, since the command accepts TCP or UDP port numbers no higher than 1023. For example:

```
ServerIron(config)# server cache-name CacheServer1 192.168.1.101
ServerIron(config-rs-CacheServer1)# port mms
ServerIron(config-rs-CacheServer1)# port pnm
ServerIron(config-rs-CacheServer1)# exit
ServerIron(config)# ip policy 1 cache tcp 0 global
ServerIron(config)# ip policy 2 cache udp 0 global
```

---

# Chapter 11

## Configuring Layer 7 Switching

This chapter describes how to configure Layer 7 switching on the ServerIron using the Command Line Interface (CLI) and Web management interface. See the *Foundry ServerIron Command Line Interface Reference* for information about the CLI commands.

Layer 7 switching allows the ServerIron to make forwarding decisions about HTTP traffic using information in a URL, cookie, or SSL session ID. The ServerIron can perform the following kinds of Layer 7 switching:

- **URL switching** directs HTTP requests to a server group using information in the text of a URL string. See “Configuring URL Switching” on page 11-1.
- **Cookie switching** directs HTTP requests to a server or server group based on information embedded in a cookie in the HTTP header. See “Configuring Cookie Switching” on page 11-22.
- **Concurrent URL and cookie switching** directs HTTP request messages to real servers based first on the contents of the Cookie header (if any) and then on the contents of the URL string. See “Using URL Switching and Cookie Switching Concurrently” on page 11-27.
- **HTTP header hashing** internally maps certain kinds of information in an HTTP header to a real server and directs all HTTP requests that contain this information to this real server. See “Configuring HTTP Header Hashing” on page 11-30.
- **SSL Session ID switching** connects a client to the same server to which it had previously established an SSL (Secure Sockets Layer) connection. See “Configuring SSL Session ID Switching” on page 11-40.

In addition, you can display information about the ServerIron's Layer 7 switching configuration, including policy definitions and Layer 7 switching statistics. See “Viewing Layer 7 Switching Details and Statistics” on page 11-47.

---

**NOTE:** You cannot use FWLB **and** the features described in this chapter on the same ServerIron.

---

### Configuring URL Switching

URL switching is the ServerIron's ability to direct HTTP requests to a server, or group of servers, using information in the text of a URL string. The ServerIron examines the contents of a URL string and makes a decision about where to send the packet based on selection criteria in user-defined policies. If text in the URL string matches the selection criteria, the HTTP request is sent to a load-balanced server group specified in the policy.

"URL string" is defined as *the contents of the Request-URI part of the Request-Line in an HTTP request message*. This information usually consists of the absolute pathname (directory and filename) of a resource. For example:

/doc/ServerIron/1199/url\_switching.html

The URL string can also be the input to a process running on a remote server. For example:

/quote.cgi?s=FDRY&d=1d

The network location of the resource is specified in the Host header field in an HTTP request message. For example:

Host: www.foundrynet.com

The ServerIron can examine both the URL string and Host header field when determining where to send the HTTP request. See RFC 1945 or RFC 2616 for more information on HTTP request messages.

The selection criteria in a policy can be a string of characters starting from the beginning of the URL string, end of the URL string, or within any part of the URL string. For example, selection criteria can be a URL string that starts with the text "/home". When an HTTP request that has a URL string beginning with the text "/home" comes into the ServerIron, the policy can direct that request to the server group containing the web content for the site's /home directory (or to another URL switching policy for additional matching).

Unlike standard server load balancing, which requires that the same content be on all load-balanced real servers, URL switching allows you to place different web content on different servers. For example, you can place image files on one group of servers and CGI applications on another group. Information in the URL string determines to which server group HTTP requests are sent.

## Basic URL Switching Example

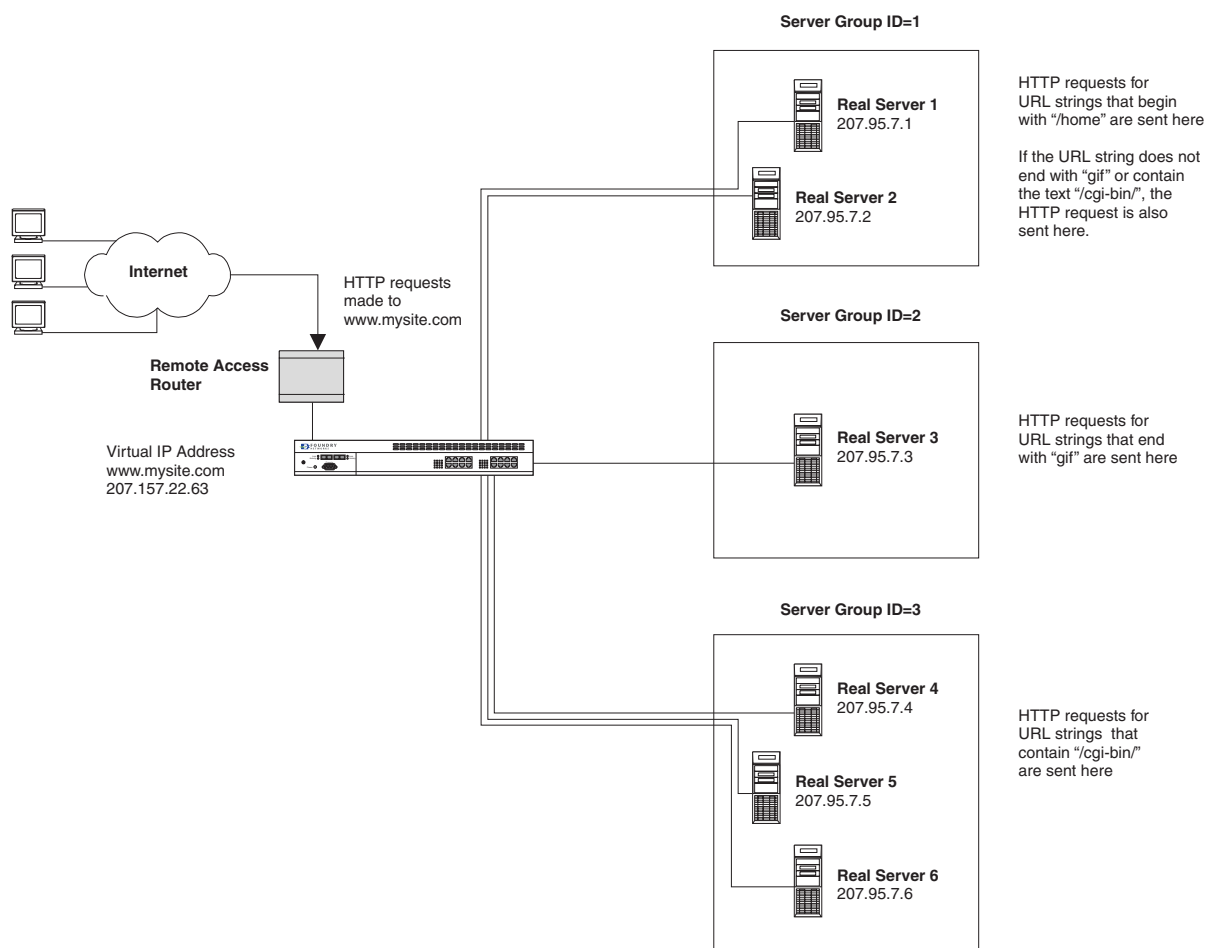
The diagram in Figure 11.1 illustrates a basic example of URL switching. The ServerIron is connected to three groups of load-balanced real servers. The server group with ID = 1 contains the /home directory for the web site. The server group with ID = 2 contains all the GIF files for the web site. The server group with ID = 3 contains all the CGI applications for the web site.

The ServerIron has URL switching policies in place that cause HTTP requests to be directed as follows:

- HTTP requests containing URL strings that start with the text "/home" are sent to server group ID = 1.
- HTTP requests containing URL strings that end with the text ".gif" are sent to server group ID = 2.
- HTTP requests containing URL strings that have the text "/cgi-bin/" anywhere within are sent to server group ID = 3.
- If a URL string does not start with the text "/home", end with the text ".gif", or contain the text "/cgi-bin/", the HTTP request is sent to server group ID = 1.



Figure 11.1 Example of a URL switching configuration



Setting up URL switching consists of the following steps:

1. Setting up the URL switching policies
2. Configuring the real servers
3. Setting up the virtual server

These tasks are described in the following sections.

### Setting Up URL Switching Policies

URL switching policies define selection criteria for URL strings and specify what happens when a URL string matches the selection criteria. If an HTTP request contains a URL string that matches a policy's selection criteria, the HTTP request can be sent to a load-balanced real server group or to another policy for additional matching.

**NOTE:** The URL switching policies discussed in this section apply to the example in Figure 11.1.

#### USING THE CLI

The following commands define a URL switching policy called p1.

```
ServerIron(config)# url-map p1
ServerIron(config-url-p1)# method prefix
ServerIron(config-url-p1)# match "/home" 1
ServerIron(config-url-p1)# default p2
```

```
ServerIron(config-url-p1)# exit
```

**Syntax:** url-map <policy-name>

**Syntax:** method prefix | suffix | pattern

**Syntax:** match "<selection-criteria>" <server-group-id> | <policy-name>

**Syntax:** default <server-group-id> | <policy-name>

The **url-map p1** command sets the name of the policy and enters the URL switching CLI level.

The **method prefix** command specifies what kind of matching the policy does on the selection criteria. Three kinds of matching methods are supported:

**prefix**        Compares the selection criteria to the beginning of the URL string.

**suffix**       Compares the selection criteria to the end of the URL string.

**pattern**      Looks for the selection criteria anywhere within the URL string.

The **match "/home" 1** command consists of two parts. The first part specifies the selection criteria, which can be up to 80 characters in length; the second part indicates what to do when the URL string matches the selection criteria – either send the HTTP request to a real server group or match the URL string against another policy. In this example, the selection criteria is the text string "/home". Since the matching method is **prefix**, the policy looks at the URL string starting from the beginning. If the URL string starts with the text "/home", then the URL string meets the selection criteria.

---

**NOTE:** In addition to using text as selection criteria, you can use an asterisk (\*) as a wildcard character to specify one or more characters at the end of a URL string. For example, using "/ho\*" as the selection criteria matches /home, /hotels, and /home/main/index.html. See the definition of policyA on page 11-14 for an example of this.

---

If the URL string meets the selection criteria, the second part of the **match** command specifies what to do with the HTTP request. In this example, the **1** in the command causes the HTTP request to be sent to the real server group whose ID = 1. A URL switching policy can contain multiple **match** commands, each with different selection criteria.

---

**NOTE:** You can also specify a URL switching policy name instead of a real server group ID. In this case, if part of the URL string matches the selection criteria, the remaining text of the URL string (that is, the text that was not matched by the selection criteria) is evaluated by the specified policy. See the definition of policyA on page 11-14 for an example of this.

---

The **default p2** command specifies what happens when the URL string does not meet any of the selection criteria in a URL switching policy's **match** command. As with a **match** command, you can specify either a real server group ID number or another URL switching policy. In this example, if a URL string does not match the selection criteria in policy p1, it is sent to policy p2 for evaluation.

The following commands define URL switching policy p2 for the example in Figure 11.1.

```
ServerIron(config)# url-map p2
ServerIron(config-url-p2)# method suffix
ServerIron(config-url-p2)# match "gif" 2
ServerIron(config-url-p2)# default p3
ServerIron(config-url-p2)# exit
```

URL switching policy p2 uses the suffix matching method. This means that the last part of the URL string is compared to the selection criteria. The **match** command defines the selection criteria as the text "gif". Thus, any URL string that ends with "gif" meets the selection criteria. The second part of the **match** command causes HTTP requests for URL strings that end in "gif" to be sent to the real server group whose ID = 2. If the URL string does not end with "gif", the **default p3** command causes the URL string to be evaluated by URL switching policy p3.

**NOTE:** As the diagram in Figure 11.1 illustrates, there is only one real server in server group ID = 2. Even so, the **match** command must refer to a server group, rather than an actual real server. Server groups can consist of one or more real servers.

The following commands define URL switching policy p3 for the example in Figure 11.1.

```
ServerIron(config)# url-map p3
ServerIron(config-url-p3)# method pattern
ServerIron(config-url-p3)# match "/cgi-bin/" 3
ServerIron(config-url-p3)# default 1
ServerIron(config-url-p3)# exit
```

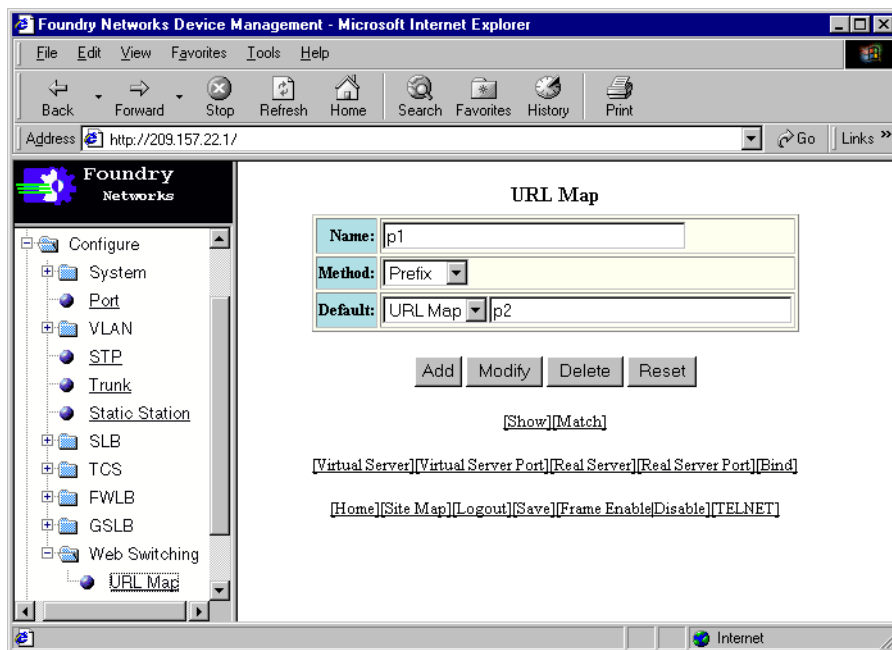
URL switching policy p3 uses the pattern matching method. The **match** command looks for the selection criteria anywhere within the URL string. In this example, if the text "/cgi-bin/" appears anywhere in the URL string, the HTTP request is sent to the real server group whose ID = 3. If "/cgi-bin/" does not appear in the URL string, the **default 1** command sends the HTTP request to the real server group whose ID = 1.

**NOTE:** If you are using the suffix matching method, you cannot use an asterisk (\*) as a wildcard character. The asterisk wildcard character is valid for the prefix and pattern matching methods only.

### USING THE WEB MANAGEMENT INTERFACE

To configure a URL switching policy using the Web Management interface:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to Web Switching in the tree view to expand the list of Layer 7 switching option links.
4. Select the URL Map link. The following panel is displayed:



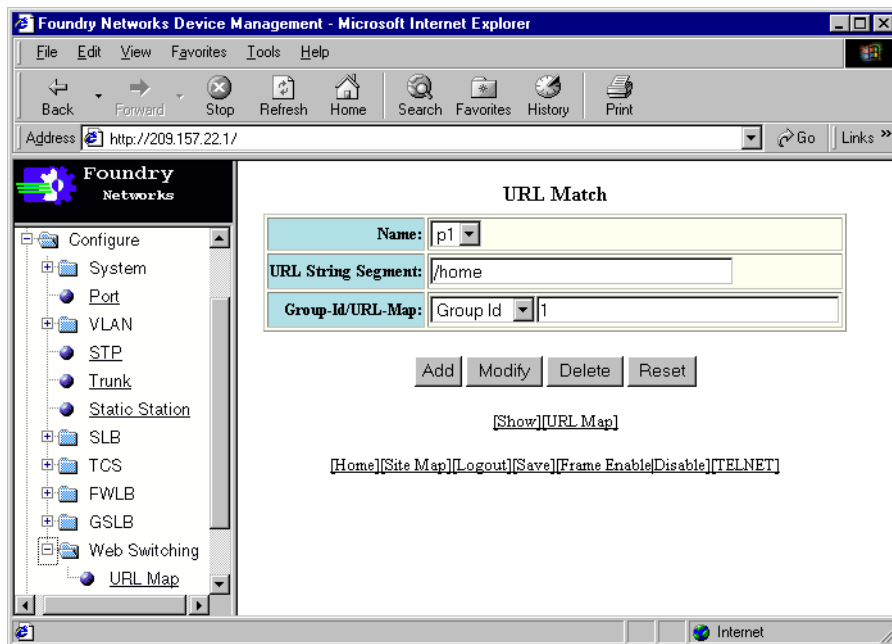
5. In the Name field, enter the name of this URL switching policy.
6. Select the matching method for this URL switching policy from the Method pulldown menu. Specify one of the following:

**Prefix** Compares the selection criteria to the beginning of the URL string.

**Suffix** Compares the selection criteria to the end of the URL string.

**Pattern** Looks for the selection criteria anywhere within the URL string.

7. In the Default field, specify what happens when the URL string does not meet any of the policy's selection criteria. The HTTP request can be directed to a real server group, or the URL string can be matched against another policy.
  - To direct the HTTP request to a real server group, select Group Id from the pulldown menu and enter the ID of the real server group.
  - To match the URL string against another URL switching policy, select URL Map from the pulldown menu and enter the name of the policy.
8. Click the Add button to add the URL switching policy to the device's running-config file.
9. Click the Match link. The following panel is displayed:



10. Select the URL switching policy from the Name pulldown menu.
11. In the URL String Segment field, enter the selection criteria for the policy. The selection criteria can be up to 80 characters in length.

**NOTE:** In addition to using text as selection criteria, you can use an asterisk (\*) as a wildcard character to specify one or more characters at the end of a URL string. For example, using "/ho\*" as the selection criteria matches /home, /hotels, and /home/main/index.html.

12. In the Group-Id/URL-Map field, specify what happens when the URL string meets the policy's selection criteria. The HTTP request can be directed to a real server group, or the URL string can be matched against another policy.
  - To direct the HTTP request to a real server group, select Group Id from the pulldown menu and enter the ID of the real server group.
  - To match the URL string against another URL switching policy, select URL Map from the pulldown menu and enter the name of the policy.
13. Click Add to save the changes to the device's running-config file.

14. Repeat Steps 11 – 13 for each set of selection criteria in the policy.
15. Select the [Save](#) link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Configuring the Real Servers

The real servers contain the web content that is returned to the requesting clients. When configuring URL switching, you place the real servers into logical server groups. URL switching policies direct HTTP requests to these logical groups, rather than to the real servers themselves.

A server group can contain one or more real servers. If there is more than one real server in a server group, HTTP requests are load balanced across all the servers in the group. You must establish the IP address of each real server in a URL switching configuration and specify the server group to which it belongs.

#### USING THE CLI

To configure real server rs1 in Figure 11.1 on page 11-3:

```
ServerIron(config)# server real-name rs1 207.95.7.1
ServerIron(config-rs-rs1)# port http group-id 1 1
ServerIron(config-rs-rs1)# exit
```

**Syntax:** server real-name <real-server-name> <ip-addr>

**Syntax:** port http group-id <server-group-id-pairs>

The **server real-name** command defines a real server called rs1 with an IP address of 207.95.7.1.

The **port http group-id** command indicates the server group(s) to which the real server belongs. The server group is expressed as a pair of numbers, indicating a range of real server group IDs. The first number is the lowest-numbered server group ID, and the second is the highest-numbered server group ID. For example, if a real server belongs only to the server group with ID = 1, the last two numbers in the **port http group-id** command would be **1 1**. (Note the space between the two numbers.) If a real server belongs to server groups 1 – 10, the last two numbers in the command would be **1 10**. Valid numbers for server group IDs are 0 – 1023.

To include a real server in groups that are not consecutively numbered, you can enter up to four server group ID pairs. For example, to include a real server in groups 1 – 5 and 11 – 15, you would enter the following command:

```
ServerIron(config-rs-rs1)# port http group-id 1 5 11 15
```

You can also specify the server group ID pairs on separate lines; for example:

```
ServerIron(config-rs-rs1)# port http group-id 1 5
ServerIron(config-rs-rs1)# port http group-id 11 15
```

The configuration for the remaining real servers in Figure 11.1 is shown below. These commands place real server rs2 in server group ID = 1 (along with real server rs1), real server rs3 in server group ID = 2, and real servers rs4, rs5, and rs6 in server group ID = 3.

```
ServerIron(config)# server real-name rs2 207.95.7.2
ServerIron(config-rs-rs2)# port http group-id 1 1
ServerIron(config-rs-rs2)# exit
```

```
ServerIron(config)# server real rs3 207.95.7.3
ServerIron(config-rs-rs3)# port http group-id 2 2
ServerIron(config-rs-rs3)# exit
```

```
ServerIron(config)# server real rs4 207.95.7.4
ServerIron(config-rs-rs4)# port http group-id 3 3
ServerIron(config-rs-rs4)# exit
```

```
ServerIron(config)# server real rs5 207.95.7.5
ServerIron(config-rs-rs5)# port http group-id 3 3
ServerIron(config-rs-rs5)# exit
```

```
ServerIron(config)# server real rs6 207.95.7.6
ServerIron(config-rs-rs6)# port http group-id 3 3
ServerIron(config-rs-rs6)# exit
```

### USING THE WEB MANAGEMENT INTERFACE

To create a real server and assign it to a server group, use the following procedure. Repeat this procedure for each real server to be used in URL switching.

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of server load balancing option links.
4. Select the Real Server link. The following panel is displayed:

The screenshot shows a web browser window titled "Foundry Networks Device Management - Microsoft Internet Explorer". The address bar shows "http://209.157.22.1/". The left sidebar contains a tree view with the following structure:

- ServerIron
  - Monitor
  - Configure
    - System
    - Port
    - VLAN
    - STP
    - Trunk
    - Static Station
    - SLB
      - General
      - Backup
      - Bind
      - Real Server**
      - Real Server Port

The main content area is titled "Real Server" and contains the following fields:

Server Name:	rs1
Server IP:	207.95.7.1
Maximum Connections:	1000000
Weight:	1
Host Range:	1
Remote:	<input type="checkbox"/>
Source NAT:	<input type="checkbox"/>

Below the fields are buttons: Add, Modify, Delete, and Reset. Below these buttons is a [Show] link. At the bottom of the panel are links: [Virtual Server][Virtual Server Port][Real Server Port][Bind], [Home][Site Map][Logout][Save][Frame Enable][Disable][TELNET].

5. In the Server Name field, enter the name of the real server.
6. In the Server IP field, enter the IP address of the real server.
7. Click the Add button to add the real server to the device's running-config file.
8. Click the Real Server Port link at the bottom of the panel. The Real Server Port panel is displayed.

9. Click the Modify button for the real server you just created. The following panel is displayed:

The screenshot shows the Foundry Networks web interface in Microsoft Internet Explorer. The address bar shows <http://209.157.22.1/>. The left sidebar contains a tree view with the following structure:

- ServerIron
  - Monitor
  - Configure
    - System
    - Port
    - VLAN
    - STP
    - Trunk
    - Static Station
    - SLB
      - General
      - Backup
      - Bind
      - Real Server
      - Real Server Port
      - Router Interface
      - Source IP
      - TCP/UDP Port
      - Virtual Server
      - Virtual Server Port
    - TCS
    - FWLB
    - GSLB
    - Web Switching
    - Command

The main panel is titled 'Real Server Port' and contains the following configuration fields:

- Server Name:** rs1
- TCP/UDP Port:** HTTP (with a 'User Define' button)
- Status:** ☐ Disable ☒ Enable
- Keep Alive:** ☐
- DNS Parameters:**
  - +DNS Zone:
  - +Addr Query:
  - +Proxy: ☐
- HTTP Parameters:**
  - \*Method: HEAD
  - \*URL:
  - \*Status Code:
- Group Id Range:**

From	To
1	1

At the bottom of the panel are buttons: Add, Modify, Delete, and Reset. Below these buttons is a link: [Show Real Server Port]. At the very bottom, there is a note: \* -> HTTP Only, +> DNS Only, and a row of links: [Virtual Server][Virtual Server Port][Real Server][Bind]. At the bottom of the page are links: [Home][Site Map][Logout][Save][Frame Enable/Disable][TELNET].

10. Select the real server from the Server Name pulldown menu.
11. Select HTTP from the TCP/UDP Port pulldown menu.
12. In the Group Id Range fields, enter the server group(s) to which the real server belongs. You specify the server group IDs in terms of one or more ranges:
  - In the From field, enter the lowest-numbered server group ID to which this real server belongs.
  - In the To field, enter the highest-numbered server group ID to which this real server belongs.

You can enter up to four ranges of server group IDs. Valid numbers for server group IDs are 0 – 1023. If the real server belongs to only one server group, you can enter the server group ID in the first From field and leave the other Group Id Range fields blank.
13. Click the Add button to save the changes to the device's running-config file.
14. Select the Save link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Setting up the Virtual Server

Once you have set up the URL switching policy and placed the real servers into groups, you bind the policy to the virtual IP address (VIP). After you do this, HTTP requests coming to the VIP are evaluated by the policies and sent to the appropriate server group.

---

**NOTE:** You cannot use an access control list on a VIP that has URL switching enabled.

---

### USING THE CLI

The following example applies to the configuration in Figure 11.1 on page 11-3:

```
ServerIron(config)# server virtual-name mysite 209.157.22.63
ServerIron(config-vs-mysite)# port http
ServerIron(config-vs-mysite)# port http url-map p1
ServerIron(config-vs-mysite)# port http url-switch
ServerIron(config-vs-mysite)# bind http rs1 http
ServerIron(config-vs-mysite)# bind http rs2 http
ServerIron(config-vs-mysite)# bind http rs3 http
ServerIron(config-vs-mysite)# bind http rs4 http
ServerIron(config-vs-mysite)# bind http rs5 http
ServerIron(config-vs-mysite)# bind http rs6 http
ServerIron(config-vs-mysite)# exit
```

**Syntax:** server virtual-name <virtual-server-name> <ip-addr>

**Syntax:** port http

**Syntax:** port http url-map <policy-name>

**Syntax:** port http url-switch

**Syntax:** bind http <real-server-name> http

The **server virtual** command defines a virtual server called mysite with an IP address of 209.157.22.63 and enters the virtual server CONFIG level for this VIP.

The **port http** command adds port 80 (HTTP) to the VIP.

The **port http url-map** command specifies a URL switching policy to be active for this VIP. If you configure more than one URL switching policy, the policies must be linked together. In this example, policy p1 may send text to policy p2, which, in turn, may send text to policy p3. Thus, the three policies are linked together. Up to 100 URL switching policies can be linked in this way.

The **port http url-switch** command activates URL switching for this VIP. You must have already defined the URL switching policies before entering this command.

The **bind http** commands bind the virtual server to HTTP services on the real servers. In this example, the commands associate real servers rs1 – rs6 with the virtual server.

---

**NOTE:** For clarity, the bindings in the example above are shown as six separate entries. Alternatively, you can enter all the binding information as one command: **bind http rs1 http rs2 http rs3 http rs4 http rs5 http rs6 http**.

---

### USING THE WEB MANAGEMENT INTERFACE

To configure a virtual server for URL switching:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of server load balancing option links.



4. Select the Virtual Server link. The following panel is displayed:

The screenshot shows a Microsoft Internet Explorer browser window at the address <http://209.157.22.1/>. The Foundry Networks logo is in the top left. A left-hand navigation tree is expanded to 'Virtual Server'. The main content area is titled 'Virtual Server' and contains a form with the following fields:

Server Name:	<input type="text" value="mysite"/>
Server IP:	<input type="text" value="209.157.22.63"/>
Host Range:	<input type="text" value="1"/>
Symmetric Priority:	<input type="text" value="0"/>
HTTP Redirect:	<input type="checkbox"/>
Load Balancing Metric:	<input checked="" type="radio"/> Default <input type="radio"/> Least Connection <input type="radio"/> Round Robin <input type="radio"/> Weighted

Below the form are four buttons: Add, Modify, Delete, and Reset. At the bottom of the panel, there are several links: [Show Virtual Server], [Track][Virtual Server Port][Real Server][Real Server Port][Bind], and [Home][Site Map][Logout][Save][Frame Enable][Disable][TELNET].

5. In the Server Name field, enter the name of the virtual server.
6. In the Server IP field, enter the IP address of the virtual server.
7. Click the Add button to add the virtual server to the device's running-config file.
8. Click the Virtual Server Port link at the bottom of the panel. The Virtual Server Port panel is displayed.

9. Click the Modify button for the virtual server you just created. The following panel is displayed:

The screenshot shows a web browser window titled "http://209.157.22.1/ - Microsoft Internet Explorer". The address bar shows "http://209.157.22.1/". The page content is from Foundry Networks. On the left is a navigation tree with the following items: ServerIron, Monitor, Configure, System, Port, VLAN, STP, Trunk, Static Station, SLB (selected), General, Backup, Bind, Real Server, Real Server Port, Router Interface, Source IP, TCP/UDP Port, Virtual Server, Virtual Server Port (selected), TCS, FWLB, GSLB, Web Switching, and Command. The main content area is titled "Virtual Server Port". It contains the following configuration fields:

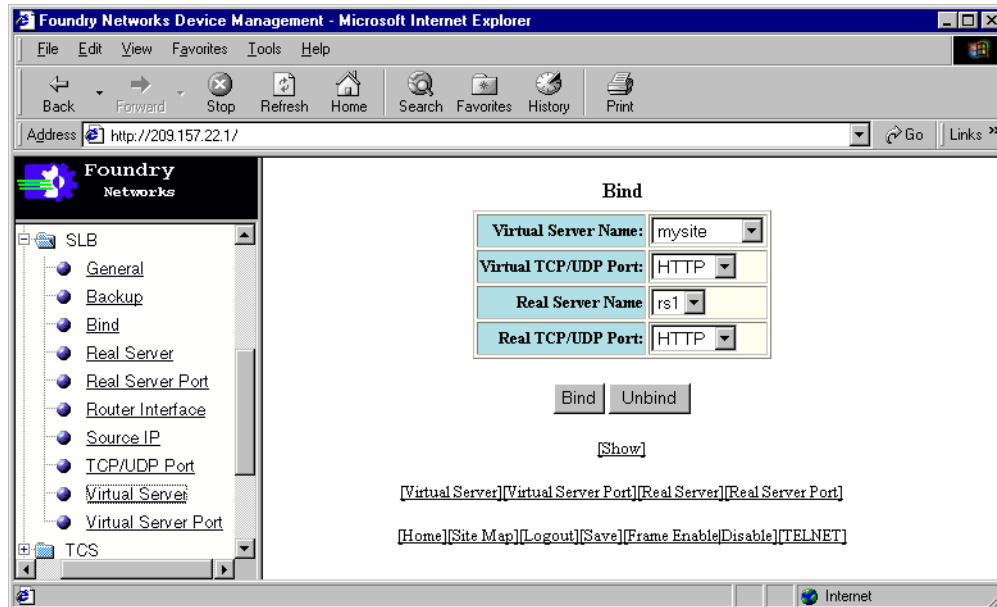
- Server Name: mysite (dropdown)
- TCP/UDP Port: HTTP (dropdown) with a "User Define" button
- Status: ☐ Disable ☒ Enable
- Sticky: ☐
- Concurrent: ☐
- Translate: ☒
- DSR: ☐
- HTTP Parameters section:
  - URL Switching: ☒
  - URL Map: p1 (dropdown)
  - URL Hashing: ☐
  - URL Segment Hashing: ☐
  - Cookie Switching: ☐
  - Cookie Name: (text field)
  - Cookie Hashing: ☐
- SSL Parameter section:
  - Session-id Switching: ☐

At the bottom of the configuration area are buttons: Add, Modify, Delete, and Reset. Below these buttons is a "[Show]" link. At the very bottom of the page are several links: [Host Id][URL Segment][URL Map][Track][Virtual Server][Real Server][Real Server Port][Bind], [Home][Site Map][Logout][Save][Frame Enable|Disable][TELNET].

10. Select the virtual server from the Server Name pulldown menu.
11. Select HTTP from the TCP/UDP Port pulldown menu.
12. Click the checkbox next to URL Switching.
13. Select a URL switching policy from the URL Map pulldown menu.
14. Click the Add button to save the changes to the device's running-config file.

**NOTE:** When you click Add, the ServerIron examines the configured URL switching policies to see that each policy called by another policy actually exists. If a policy is called but does not exist, the ServerIron displays a message indicating the URL map is not valid. If you see this message, make sure the URL switching policies are configured correctly.

15. Click the Bind link at the bottom of the panel. The Bind panel is displayed.



16. Select the virtual server from the Virtual Server Name pulldown menu.
17. Select HTTP from the Virtual TCP/UDP Port pulldown menu.
18. From the Real Server Name pulldown menu, select a real server that you are binding to this virtual server.
19. Click Bind to bind the real server to the virtual server.
20. Repeat Steps 17 – 19 for each real server you want to bind to this virtual server.
21. Select the Save link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## URL Switching Example for Two Web Sites Using One VIP

Figure 11.2 on page 11-14 illustrates another example of a URL switching configuration. This example demonstrates the following features of URL switching:

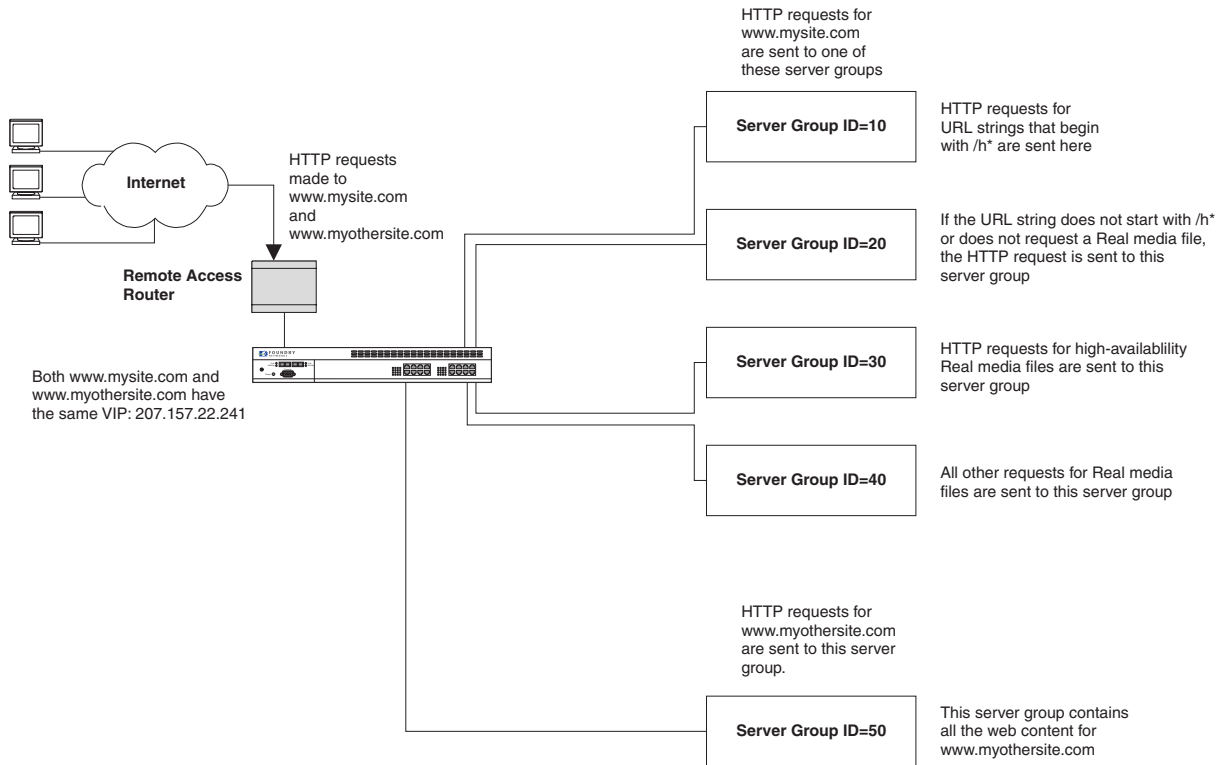
- How the ServerIron can examine the Host header field in addition to the URL string when determining where to send an HTTP request
- How you can use wildcards as selection criteria in **match** commands
- How a URL switching policy can pass a URL string to another policy for additional evaluation

In this configuration, two web sites, [www.mysite.com](http://www.mysite.com) and [www.myothersite.com](http://www.myothersite.com), share the same virtual IP address. HTTP requests for either of these web sites are sent to one of five server groups, depending on the content of the URL string.

- Requests for [www.mysite.com](http://www.mysite.com) that have URL strings beginning with the text `/h` are sent to server group ID = 10
- Requests for Real media files at [www.mysite.com](http://www.mysite.com) are sent to either server group ID = 30 (for high availability files) or server group ID = 40 (for all other Real media files)
- All other HTTP requests for [www.mysite.com](http://www.mysite.com) are sent to server group ID = 20
- All HTTP requests for [www.myothersite.com](http://www.myothersite.com) are sent to server group ID = 50
- Requests sent to the VIP, but not to either [www.mysite.com](http://www.mysite.com) or [www.myothersite.com](http://www.myothersite.com) are sent to server group

ID = 10

**Figure 11.2 URL switching example with two web sites and one VIP**



**NOTE:** For clarity, the individual real servers are not depicted in the illustration. The setup procedure for the real servers is the same as the one described in “Configuring the Real Servers” on page 11-7.

The following sections explain how to set up this configuration.

### Defining the Policies

To implement the configuration in Figure 11.2, you would create three URL switching policies:

- The first two policies, policyA and policyB, apply to HTTP requests sent to www.mysite.com, as well as to HTTP requests not specifically sent to either www.mysite.com or myothersite.com
- The third policy, policyZ, applies to HTTP requests sent to www.myothersite.com

After you set up the virtual server, as described in the next section, policyA and policyB will encounter HTTP requests for www.mysite.com and requests directed to neither www.mysite.com nor myothersite.com. PolicyZ will encounter only HTTP requests for www.myothersite.com.

### USING THE CLI

The following commands define policyA:

```
ServerIron(config)# url-map policyA
ServerIron(config-url-policyA)# method prefix
ServerIron(config-url-policyA)# match /h* 10
ServerIron(config-url-policyA)# match "/real/" policyB
ServerIron(config-url-policyA)# default 20
ServerIron(config-url-policyA)# exit
```

The **method prefix** command causes the policy to examine the first part of the URL string.

The **match /h\* 10** command looks for URL strings that start with the text "/h"; for example, /home/main/index.html or /hardware/images/toolbar.gif. These HTTP requests are sent to server group ID = 10.

The **match "/real/" policyB** command causes URL strings that start with "/real/" to be evaluated by policyB. Note that rather than sending the entire URL string, policyA sends to policyB only the text that was not matched by the selection criteria.

For example, consider a URL string of "/real/high-avail/video1.ram". The first part of the string matches the selection criteria. The remaining text, "high-avail/video1.ram", is passed to policyB for evaluation.

---

**NOTE:** If the matching method in the policy is **pattern**, the entire contents of the string are passed to the policy, rather than just part of the string.

---

The **default 20** command sends HTTP requests that do not meet the selection criteria in either of the **match** commands to server group ID = 20.

The following commands define policyB:

```
ServerIron(config)# url-map policyB
ServerIron(config-url-policyB)# method prefix
ServerIron(config-url-policyB)# match "high-avail/" 30
ServerIron(config-url-policyB)# default 40
ServerIron(config-url-policyB)# exit
```

As with policyA, the **method prefix** command causes the policy to examine the first part of the URL string that is passed to it.

The **match "high-avail/" 30** command looks for the text "high-avail/" at the beginning of the string passed to it. In this configuration, policyA sends text to policyB. (Up to 100 URL switching policies can be linked in this way.) Thus, for a URL string of "/real/high-avail/video1.ram", policyA would match the text "/real/" and pass "high-avail/video1.ram" to policyB. The text "high-avail/" would then be the first part of the string received by policyB, and would meet the selection criteria in policyB's **match** command.

The second part of policyB's **match** command sends HTTP requests meeting the selection criteria to server group ID = 30.

The **default 40** command sends HTTP requests that do not meet the selection criteria in the **match** command to server group ID = 40. This means that an HTTP request containing a URL string that starts with "/real" (but not "/real/high-avail") would go to server group ID = 40.

The following commands define policyZ:

```
ServerIron(config)# url-map policyZ
ServerIron(config-url-policyZ)# default 50
ServerIron(config-url-policyZ)# exit
```

This policy simply sends all HTTP requests it encounters to server group ID = 50. In the sample configuration in Figure 11.2 on page 11-14 all the web content for www.myothersite.com resides on the real servers in server group ID = 50.

#### USING THE WEB MANAGEMENT INTERFACE

See page 11-5 for a description of how to create a URL switching policy using the Web Management interface.

#### Setting up the Virtual Server

The two web sites in Figure 11.2 on page 11-14, www.mysite.com and www.myothersite.com, share virtual IP address 209.157.22.241. HTTP requests for either of these sites go to this one VIP. Since the URL string refers to a directory and a file, not to a host, you cannot tell from the URL string which site the HTTP request is for.

The Host header field in an HTTP request message refers to the site being requested. You can configure the ServerIron to look at the Host header field and activate a URL switching policy when a specified host is encountered.

### USING THE CLI

The **port http url-host-id** command specifies the host that the ServerIron looks for in the Host header field in an HTTP request message. The following commands apply to the virtual server in Figure 11.2 on page 11-14.

```
ServerIron(config)# server virtual sharedVIP 209.157.22.241
ServerIron(config-vs-sharedVIP)# port http
ServerIron(config-vs-sharedVIP)# port http url-host-id www.mysite.com policyA
ServerIron(config-vs-sharedVIP)# port http url-host-id www.myothersite.com policyZ
ServerIron(config-vs-sharedVIP)# port http url-map policyA
ServerIron(config-vs-sharedVIP)# port http url-switch
ServerIron(config-vs-sharedVIP)# bind http real-server1 http (other real servers...)
ServerIron(config-vs-sharedVIP)# exit
```

**Syntax:** port http url-host-id <host> <policy-name>

The **port http url-host-id www.mysite.com policyA** command causes HTTP requests for www.mysite.com to be evaluated by policyA.

The **port http url-host-id www.myothersite.com policyZ** command causes HTTP requests for www.myothersite.com to be evaluated by policyZ.

If a request is for neither www.mysite.com nor www.myothersite.com, then the request is evaluated by policyA. In this example, the **port http url-map policyA** command functions similarly to the **default** command in a URL switching policy, sending requests that don't meet the other selection criteria to a "catch-all" policy.

### Using a Wildcard Character in the port http url-host-id Command

You can use an asterisk (\*) as a wildcard character to specify one or more characters at the beginning of the Host header field. For example, specifying "\*.com" as the <host> in the **port http url-host-id** command matches all requests for hosts ending with .com. The following commands illustrate the use of the wildcard character in the **port http url-host-id** command.

```
ServerIron(config)# server virtual sharedVIP 209.157.22.241
ServerIron(config-vs-sharedVIP)# port http
ServerIron(config-vs-sharedVIP)# port http url-host-id *.com policyA
ServerIron(config-vs-sharedVIP)# port http url-host-id www.myothersite.com policyZ
ServerIron(config-vs-sharedVIP)# port http url-map policyA
ServerIron(config-vs-sharedVIP)# port http url-switch
ServerIron(config-vs-sharedVIP)# bind http real-server1 http (other real servers...)
ServerIron(config-vs-sharedVIP)# exit
```

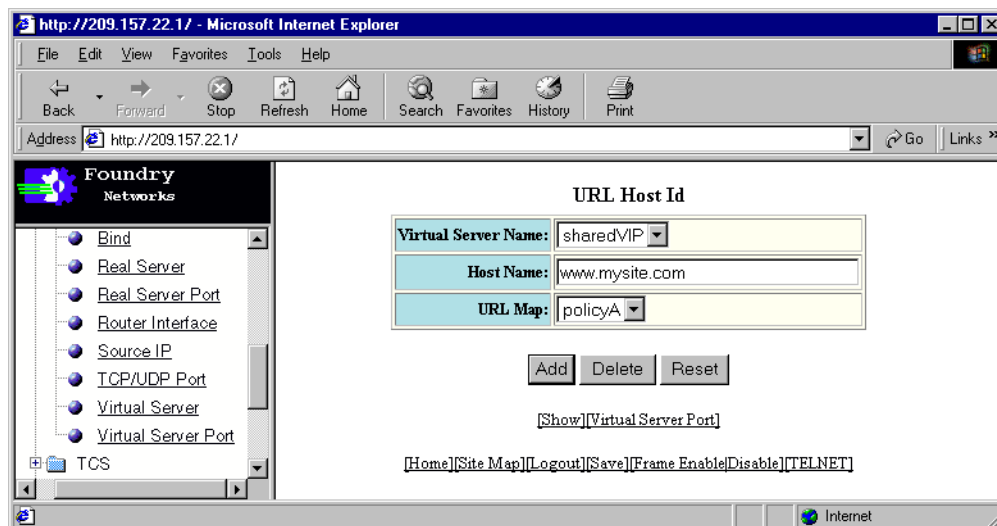
In this configuration, the **port http url-host-id \*.com policyA** command causes HTTP requests for any site ending in .com to be evaluated by policyA. Note that when there are multiple **port http url-host-id** commands in a virtual server's configuration, the ServerIron favors an exact match over a wildcard match. In the sample configuration above, any requests for www.myothersite.com are evaluated by policyZ, not policyA.

### USING THE WEB MANAGEMENT INTERFACE

To configure a virtual server to look at the Host header field in HTTP requests and activate a URL switching policy when a specified host is encountered, use the following procedure:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of server load balancing option links.
4. Select the Virtual Server link to display the Virtual Server panel.
5. In the Server Name field, enter the name of the virtual server.
6. In the Server IP field, enter the IP address of the virtual server.
7. Click the Add button to add the virtual server to the device's running-config file.
8. Click the Virtual Server Port link at the bottom of the panel to display the Virtual Server Port panel.
9. Click the Modify button for the virtual server you just created.

10. Select the virtual server from the Server Name pulldown menu.
11. Select HTTP from the TCP/UDP Port pulldown menu.
12. Click the checkbox next to URL Switching.
13. Select a URL switching policy from the URL Map pulldown menu.
14. Click the Host Id link at the bottom of the panel. The following panel is displayed:



15. Select the virtual server from the Server Name pulldown menu.
16. In the Host Name field, enter a host name. The ServerIron will look for this host name in the Host header field in HTTP requests.
17. Select a URL switching policy from the URL Map pulldown menu. The ServerIron will apply this policy when it encounters an HTTP request that contains the specified host name.
18. Click the Add button to save the changes to the device's running-config file.
19. Repeat Steps 16 – 18 for each host name you want to associate with a URL switching policy.
20. Click the Add button to save the changes to the device's running-config file.
21. Click the Virtual Server Port link at the bottom of the panel. The Virtual Server panel is displayed.
22. Click the Bind link at the bottom of the panel. The Bind panel is displayed.
23. Select the virtual server from the Virtual Server Name pulldown menu.
24. Select HTTP from the Virtual TCP/UDP Port pulldown menu.
25. From the Real Server Name pulldown menu, select a real server that you are binding to this virtual server.
26. Click Bind to bind the real server to the virtual server.
27. Repeat Steps 24 – 26 for each real server you want to bind to this virtual server.
28. Select the Save link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Sample URLs

Using the configuration in Figure 11.2 on page 11-14, URL switching policies would direct HTTP requests as follows:

**www.mysite.com/home/index.html**

The **port http url-host-id** command in the virtual server configuration directs HTTP requests for `www.mysite.com` to policyA. A match command in policyA specifies that URLs that begin with `/h/` go to server group 10.

#### **www.mysite.com/marketing**

Since the requested host is `www.mysite.com`, the HTTP request is evaluated by policyA. The URL string `/marketing` does not match any of the selection criteria in policyA's **match** commands. The **default** command sends the request to server group 20.

#### **www.mysite.com/real/high-avail/bigvideo.ram**

Since the requested host is `www.mysite.com`, the HTTP request is evaluated by policyA. A match command in policyA specifies that URL strings that begin with `/real/` be evaluated by policyB. The text `"high-avail/bigvideo.ram"` is passed to policyB. A match command in policyB specifies that strings that begin with `"high-avail/"` go to server group 30.

#### **www.mysite.com/real/oldstuff/clinton.ram**

Since the requested host is `www.mysite.com`, the HTTP request is evaluated by policyA. A match command in policyA specifies that URL strings that begin with `/real/` be evaluated by policyB. The text `"oldstuff/clinton.ram"` is passed to policyB. Since the text `"oldstuff/clinton.ram"` does not match the selection criteria in policyB's **match** command, the **default** command sends the request to server group 40.

#### **www.myothersite.com/home.html**

The **port http url-host-id** command in the virtual server configuration directs HTTP requests for `www.myothersite.com` to policyZ. The **default** command in policyZ sends all HTTP requests to server group 50.

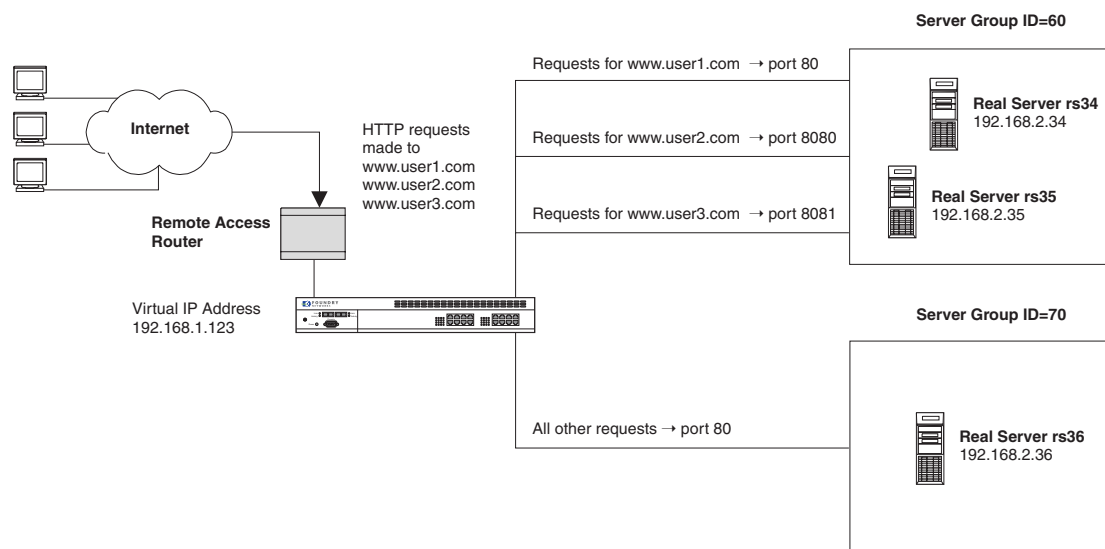
#### **209.157.22.241/home.html**

In this case, the IP address of the VIP is being requested, rather than a specific host. The **port http url-map** command in the virtual server configuration directs HTTP requests for neither `www.mysite.com` nor `www.myothersite.com` to policyA. A match command in policyA specifies that URLs that begin with `/h/` go to server group 10.

## Using URL Switching to Direct HTTP Requests to Specific TCP Ports

In addition to directing HTTP requests to real servers in server groups, URL switching allows you to specify which TCP port on the real servers in the server groups receive the HTTP requests. For example, you can configure a URL switching policy to send HTTP requests to TCP port 8080 rather than port 80. Figure 11.3 shows an example of this kind of configuration.

**Figure 11.3 Using URL switching to direct HTTP requests to specific TCP ports in a server group**





In this configuration, three domains, `www.user1.com`, `www.user2.com`, and `www.user3.com` share virtual IP address `192.168.1.123`. HTTP requests sent to `www.user1.com` go to port 80 on one of the load-balanced real servers in server group ID=60. Requests for `www.user2.com` go to port 8080, and requests for `www.user3.com` go to port 8081. Requests coming into the VIP that are addressed to none of these three domains are sent to port 80 on the real server in server group ID=70.

The following sections explain how to set up this configuration.

### Defining the Policies

To implement the configuration in Figure 11.3, you would enter the following commands to create four URL switching policies:

```
ServerIron(config)# url-map urlmap1
ServerIron(config-url-urlmap1)# tcp-port 80
ServerIron(config-url-urlmap1)# default 60
ServerIron(config-url-urlmap1)# exit

ServerIron(config)# url-map urlmap2
ServerIron(config-url-urlmap2)# tcp-port 8080
ServerIron(config-url-urlmap2)# default 60
ServerIron(config-url-urlmap2)# exit

ServerIron(config)# url-map urlmap3
ServerIron(config-url-urlmap3)# tcp-port 8081
ServerIron(config-url-urlmap3)# default 60
ServerIron(config-url-urlmap3)# exit

ServerIron(config)# url-map urlmap4
ServerIron(config-url-urlmap4)# default 70
ServerIron(config-url-urlmap4)# exit
```

**Syntax:** `tcp-port <port-number>`

In the URL switching policies above, the **tcp-port** commands specify the TCP port where HTTP requests evaluated by the policy are sent. The **default** commands specify the server group to which the HTTP requests are sent. HTTP requests that are evaluated by policy `urlmap1` are sent to TCP port 80 on one of the load-balanced real servers in server group ID = 60; requests evaluated by policy `urlmap2` are sent to TCP port 8080 on a server in server group ID = 60; and requests evaluated by policy `urlmap3` are sent to TCP port 8081 on a server in server group ID = 60. If an HTTP request is evaluated by policy `urlmap4`, it is sent to TCP port 80 (the default port) on the server in server group ID = 70.

### Configuring the Real Servers

The following commands configure the three real servers in Figure 11.3.

```
ServerIron(config)# server real-name rs34 192.168.2.34
ServerIron(config-rs-rs34)# port http group-id 60 60
ServerIron(config-rs-rs34)# exit

ServerIron(config)# server real-name rs35 192.168.2.35
ServerIron(config-rs-rs35)# port http group-id 60 60
ServerIron(config-rs-rs35)# exit

ServerIron(config)# server real-name rs36 192.168.2.36
ServerIron(config-rs-rs36)# port http group-id 70 70
ServerIron(config-rs-rs36)# exit
```

These commands place real servers `rs34` and `rs35` in server group ID = 60, and real server `rs36` in server group ID = 70.

### Setting up the Virtual Server

The following commands configure the VIP shown in Figure 11.3.

```
ServerIron(config)# server virtual vs1 192.168.1.123
ServerIron(config-vs-vs1)# port http
```

```
ServerIron(config-vs-vs1)# port http url-host-id www.user1.com urlmap1
ServerIron(config-vs-vs1)# port http url-host-id www.user2.com urlmap2
ServerIron(config-vs-vs1)# port http url-host-id www.user3.com urlmap3
ServerIron(config-vs-vs1)# port http url-map urlmap4
ServerIron(config-vs-vs1)# port http url-switch
ServerIron(config-vs-vs1)# bind http rs34 http s35 http s36 http
ServerIron(config-vs-vs1)# exit
```

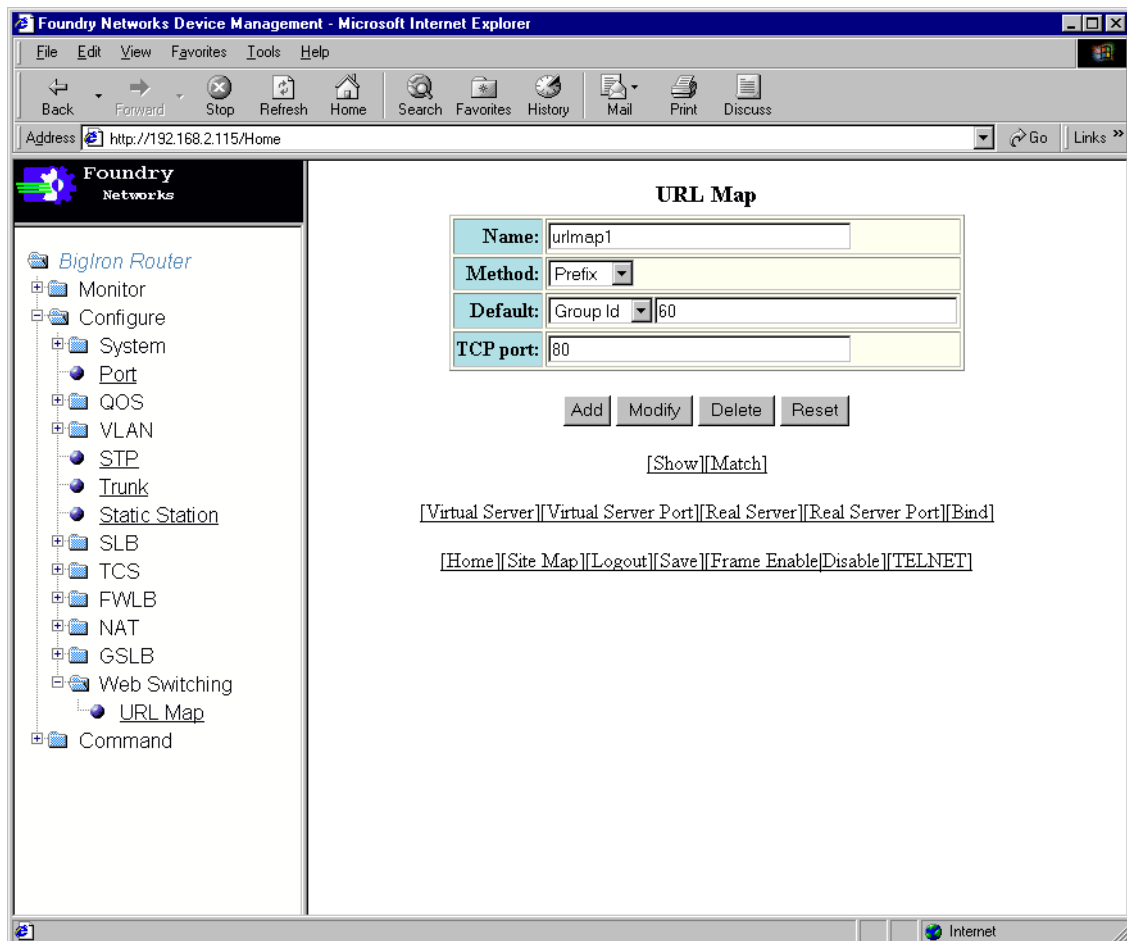
The **port http url-host-id** commands cause HTTP requests for [www.user1.com](http://www.user1.com) to be evaluated by policy [urlmap1](#), requests for [www.user2.com](http://www.user2.com) to be evaluated by policy [urlmap2](#), and requests for [www.user3.com](http://www.user3.com) to be evaluated by policy [urlmap3](#).

If a request comes into the VIP that is not for [www.user1.com](http://www.user1.com), [www.user2.com](http://www.user2.com), or [www.user3.com](http://www.user3.com), then the request is evaluated by policy [urlmap4](#).

### USING THE WEB MANAGEMENT INTERFACE

To implement the configuration in Figure 11.3, use the following procedure.

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to Web Switching in the tree view to expand the list of Layer 7 switching option links.
4. Select the [URL Map](#) link. The following panel is displayed:



5. In the Name field, enter the name of this URL switching policy; for example, urlmap1.
6. Select the matching method for this URL switching policy from the Method pulldown menu. For the policies in the configuration in Figure 11.3, you can leave the default method of Prefix.
7. In the Default field, specify what happens when the URL string does not meet any of the policy's selection criteria. For policy urlmap1, used in Figure 11.3, you would select Group Id from the pulldown menu and enter the ID of the real server group 60.
8. In the TCP port field, specify the TCP port where HTTP requests evaluated by the policy are sent. For policy urlmap1, you would enter 80, which would cause HTTP requests evaluated by the policy to be sent to TCP port 80 on one of the load-balanced real servers in server group ID = 60.
9. Click the Add button to add the URL switching policy to the device's running-config file.
10. To implement the configuration in Figure 11.3, repeat steps 5 through 9 for policies urlmap2, urlmap3, and urlmap4. When the policies are activated, requests evaluated by policy urlmap2 are sent to TCP port 8080 on a server in server group ID = 60, requests evaluated by policy urlmap3 are sent to TCP port 8081 on a server in server group ID = 60, and requests evaluated by policy urlmap4 are sent to TCP port 80 on the server in server group ID = 70.
11. Select the [Real Server](#) link at the bottom of panel to display the Real Server panel.
12. In the Server Name field, enter the name of the real server.
13. In the Server IP field, enter the IP address of the real server.
14. Click the Add button to add the real server to the device's running-config file.
15. Click the [Real Server Port](#) link at the bottom of the panel. The Real Server Port panel is displayed.
16. Click the Modify button for the real server you just created.
17. Select the real server from the Server Name pulldown menu.
18. Select HTTP from the TCP/UDP Port pulldown menu.
19. In the Group Id Range fields, enter the server group(s) to which the real server belongs. You specify the server group IDs in terms of one or more ranges:
  - In the From field, enter the lowest-numbered server group ID to which this real server belongs.
  - In the To field, enter the highest-numbered server group ID to which this real server belongs.
20. Click the Modify button to save the changes to the device's running-config file.
21. Repeat steps 11 through 20 for each real server in the configuration.
22. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
23. Click on the plus sign next to SLB in the tree view to expand the list of server load balancing option links.
24. Select the [Virtual Server](#) link to display the Virtual Server panel.
25. In the Server Name field, enter the name of the virtual server; for example, vs1
26. In the Server IP field, enter the IP address of the virtual server.
27. Click the Add button to add the virtual server to the device's running-config file.
28. Click the [Virtual Server Port](#) link at the bottom of the panel to display the Virtual Server Port panel.
29. Click the Modify button for the virtual server you just created.
30. Select the virtual server from the Server Name pulldown menu.
31. Select HTTP from the TCP/UDP Port pulldown menu.
32. Click the checkbox next to URL Switching.
33. Select a URL switching policy from the URL Map pulldown menu.
34. Click the [Host Id](#) link at the bottom of the panel to display the URL Host Id panel.

35. Select the virtual server from the Virtual Server Name pulldown menu.
36. In the Host Name field, enter a host name. The ServerIron will look for this host name in the Host header field in HTTP requests.
37. Select a URL switching policy from the URL Map pulldown menu. The ServerIron will apply this policy when it encounters an HTTP request that contains the specified host name.
38. Click the Add button to save the changes to the device's running-config file.
39. Repeat Steps 36 – 38 for each host name you want to associate with a URL switching policy.
40. Click the Add button to save the changes to the device's running-config file.
41. Click the [Virtual Server Port](#) link at the bottom of the panel. The Virtual Server panel is displayed.
42. Click the [Bind](#) link at the bottom of the panel. The Bind panel is displayed.
43. Select the virtual server from the Virtual Server Name pulldown menu.
44. Select HTTP from the Virtual TCP/UDP Port pulldown menu.
45. From the Real Server Name pulldown menu, select a real server that you are binding to this virtual server.
46. Click Bind to bind the real server to the virtual server.
47. Repeat Steps 6 – 7 for each real server you want to bind to this virtual server.
48. Select the [Save](#) link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Configuring Cookie Switching

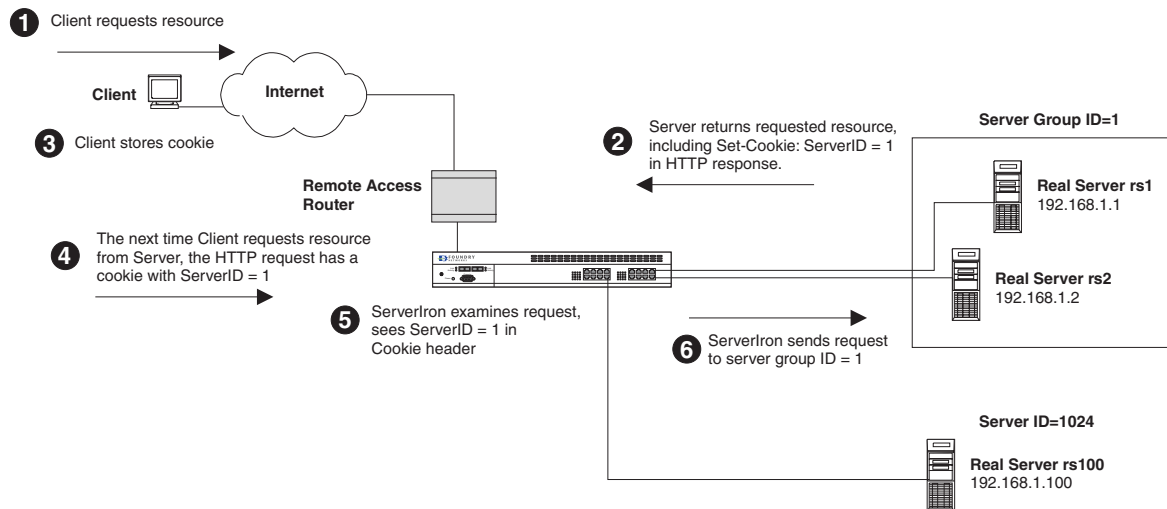
Cookie switching is the ServerIron's ability to direct HTTP requests to a server or server group based on information embedded in a cookie in the HTTP header. You configure your server to send a cookie when responding to a request from a client. The client stores the cookie, and the next time the client requests information from the server, the cookie specifies which server or server group should handle the request. In this way, you can ensure that requests from a particular client are always handled by a particular server or server group, even across sessions.

---

**NOTE:** For information on configuring the ServerIron to use cookie switching and URL switching at the same time, see "Using URL Switching and Cookie Switching Concurrently" on page 11-27.

---

The configuration in Figure 11.4 illustrates how cookie switching works.

**Figure 11.4 How cookie switching works**

This is what's going on in the diagram:

1. The client requests a resource from the server. Using its load-balancing metric, the ServerIron sends the HTTP request to one of the servers bound to the VIP.
2. The server responds to the request. The server's HTTP response contains a Set-Cookie header that includes a NAME=VALUE pair relevant to cookie switching. The NAME part is a user-defined token (for example, ServerID) that identifies the cookie; the VALUE part refers to the ID of either a server group consisting of one or more load-balanced real servers, or a specific real server.
3. The NAME=VALUE pair in the Set-Cookie header is stored on the client.
4. The next time the client requests a resource from the server, the HTTP request contains a Cookie header that includes the NAME=VALUE pair.
5. The ServerIron examines the Cookie header in the HTTP request, looking for the name of the cookie; that is, the NAME part of the NAME=VALUE pair.
6. If the cookie is found, the ServerIron directs the HTTP request to the server or server group whose ID is specified in the VALUE part of the NAME=VALUE pair.

Cookie switching provides a function similar to the "sticky" port feature in Server Load Balancing. Both features cause sequential HTTP requests from a client to go to the same server (or in the case of cookie switching, to a load-balanced server group). The difference is in how sessions are handled:

- In Server Load Balancing, when you configure port 80 (HTTP) on a virtual server to be sticky, HTTP requests from a client always go to the same real server until the session times out (that is, the sticky age timer expires). When the session times out, the ServerIron uses a load balancing metric to select a new real server to handle requests from the client. If a user's transaction is not complete when the session times out, it may be load-balanced to a new server, and the user may have to log in again.
- Cookie switching, in contrast, works independently of session. HTTP requests from a client always go to the server (that is, the real server or a server group) specified in the cookie, regardless of the interval between requests. Requests are never load balanced to a different server.

Setting up cookie switching consists of the following tasks:

1. Setting up the servers
2. Configuring the server to send a Set-Cookie header that contains a NAME=VALUE pair relevant to cookie switching
3. Enabling cookie switching on the virtual server

## Setting Up the Servers

Using information stored in a Cookie header, the ServerIron can direct an HTTP request to one of the following:

- A server group consisting of one or more load-balanced real servers
- A specific real server

The illustration in Figure 11.4 on page 11-23 shows a configuration with one real server and one server group consisting of two real servers. The following commands configure the two real servers in the server group in Figure 11.4:

```
ServerIron(config)# server real-name rs1 192.168.1.1
ServerIron(config-rs-rs1)# port http group-id 1 1
ServerIron(config-rs-rs1)# exit

ServerIron(config)# server real-name rs2 192.168.1.2
ServerIron(config-rs-rs2)# port http group-id 1 1
ServerIron(config-rs-rs2)# exit
```

**Syntax:** server real-name <real-server-name> <ip-addr>

**Syntax:** port http group-id <server-group-id-pairs>

The **port http group-id** commands indicate the server group to which the real servers belong. The server group is expressed as a pair of numbers, indicating a range of real server group IDs. The first number is the lowest-numbered server group ID, and the second is the highest-numbered server group ID. In this example, both real servers belong only to the server group with ID = 1, so the last two numbers in the **port http group-id** commands are **1 1** (note the space between the two numbers). Valid numbers for server group IDs are 0 – 1023. See “Configuring the Real Servers” on page 11-7 for more information on setting up server groups.

To direct HTTP requests to a specific real server, you can either configure a real server be the only member of a server group, or you can assign an ID to a real server. If you assign an ID to a real server, the ServerIron directs HTTP requests that contain the server’s ID value in the Cookie header to that real server.

For example, the following commands assign a server ID to real server rs100 in Figure 11.4:

```
ServerIron(config)# server real-name rs100 192.168.1.100
ServerIron(config-rs-rs100)# port http server-id 1024
ServerIron(config-rs-rs100)# exit
```

**Syntax:** port http server-id <server-id>

The **port http server-id** command sets the server ID for the real server at 1024. HTTP requests coming into the ServerIron that have a cookie value that refers to this server ID are always sent to this real server. Valid numbers for server IDs are 1024 – 2047.

## Configuring the Server to Set a Cookie

In cookie switching, you configure your server to include a Set-Cookie header in its responses to HTTP requests. This Set-Cookie header must include a NAME=VALUE pair relevant to cookie switching.

**NAME** Is a token that identifies the cookie. This can be any word (for example, ServerID), but cannot start with a dollar sign (\$).

**VALUE** Refers to the ID of a server group (0 – 1023) or a real server (1024 – 2047). See “Setting Up the Servers” on page 11-24 for information on setting up IDs for real servers and server groups.

The exact procedure for configuring a server to send a Set-Cookie header depends on your server configuration and is not discussed here. However, the following is an example of a JavaScript command to create a cookie whose NAME is ServerID and VALUE is 1:

```
SetCookie("ServerID","1")
```

Consult RFC 2109 or your JavaScript documentation for more information on the Set-Cookie header.

## Enabling Cookie Switching on the Virtual Server

To enable cookie switching on the virtual server, use one of the following methods.

---

**NOTE:** You cannot use an access control list on a VIP that has cookie switching enabled.

---

### USING THE CLI

The following commands enable cookie switching on a virtual server called cookieVIP:

```
ServerIron(config)# server virtual cookieVIP 192.168.1.241
ServerIron(config-vs-cookieVIP)# port http
ServerIron(config-vs-cookieVIP)# port http cookie-name ServerID
ServerIron(config-vs-cookieVIP)# port http cookie-switching
ServerIron(config-vs-cookieVIP)# bind http rs1 http rs2 http rs100 http
ServerIron(config-vs-cookieVIP)# exit
```

**Syntax:** port http cookie-name <name>

**Syntax:** port http cookie-switching

The **port http cookie-name** command specifies the name of the cookie to be used in cookie switching. This must be the same as the NAME token you configured your server to include in responses to HTTP requests. In the example, the cookie name is “ServerID”.

The **port http cookie-switching** command enables cookie switching on this virtual server. In this example, when the ServerIron encounters an HTTP request sent to this VIP, it looks in the Cookie header for a cookie with ServerID as the NAME part of the NAME=VALUE pair. If the cookie is found, the request is directed to the real server or the server group whose ID is specified in the VALUE part of the NAME=VALUE pair.

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of server load balancing option links.
4. Select the Virtual Server link to display the Virtual Server panel.
5. In the Server Name field, enter the name of the virtual server.
6. In the Server IP field, enter the IP address of the virtual server.
7. Click the Add button to add the virtual server to the device's running-config file.
8. Click the Virtual Server Port link at the bottom of the panel to display the Virtual Server Port panel.

9. Click the Modify button for the virtual server you just created. The following panel is displayed:

The screenshot shows a web browser window titled "http://209.157.22.1/ - Microsoft Internet Explorer". The address bar shows "http://209.157.22.1/". The browser displays the Foundry Networks configuration interface. On the left is a navigation tree with the following items: ServerIron, Monitor, Configure, System, Port, VLAN, STP, Trunk, Static Station, SLB (selected), General, Backup, Bind, Real Server, Real Server Port, Router Interface, Source IP, TCP/UDP Port, Virtual Server, Virtual Server Port, TCS, FWLB, GSLB, Web Switching, and Command. The main content area is titled "Virtual Server Port" and contains the following configuration fields:

- Server Name: cookieVIP
- TCP/UDP Port: HTTP (dropdown menu) with a "User Define" button
- Status: ☐ Disable ☒ Enable
- Sticky: ☐
- Concurrent: ☐
- Translate: ☒
- DSR: ☐
- HTTP Parameters** (section header)
- URL Switching: ☐
- URL Map: None (dropdown menu)
- URL Hashing: ☐
- URL Segment Hashing: ☐
- Cookie Switching: ☒
- Cookie Name: ServerGroup
- Cookie Hashing: ☐
- SSL Parameter** (section header)
- Session-id Switching: ☐

At the bottom of the configuration area are buttons: Add, Modify, Delete, and Reset. Below these buttons is a "[Show]" link. At the very bottom of the panel are two rows of links: "[Host Id][URL Segment][URL Map][Track][Virtual Server][Real Server][Real Server Port][Bind]" and "[Home][Site Map][Logout][Save][Frame Enable][Disable][TELNET]". The browser's status bar at the bottom shows "http://209.157.22.1/L4VSPpg.htm" and "Internet".

10. Click the checkbox next to Cookie Switching.
11. In the Cookie Name field, enter the name of the cookie to be used in cookie switching. This must be the same as the NAME token you configured your server to include in responses to HTTP requests.
12. Click the Bind link at the bottom of the panel. The Bind panel is displayed.
13. Select the virtual server from the Virtual Server Name pulldown menu.
14. Select HTTP from the Virtual TCP/UDP Port pulldown menu.
15. From the Real Server Name pulldown menu, select a real server that you are binding to this virtual server.
16. Click Bind to bind the real server to the virtual server.
17. Repeat Steps 15 – 16 for each real server you want to bind to this virtual server.
18. Select the Save link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

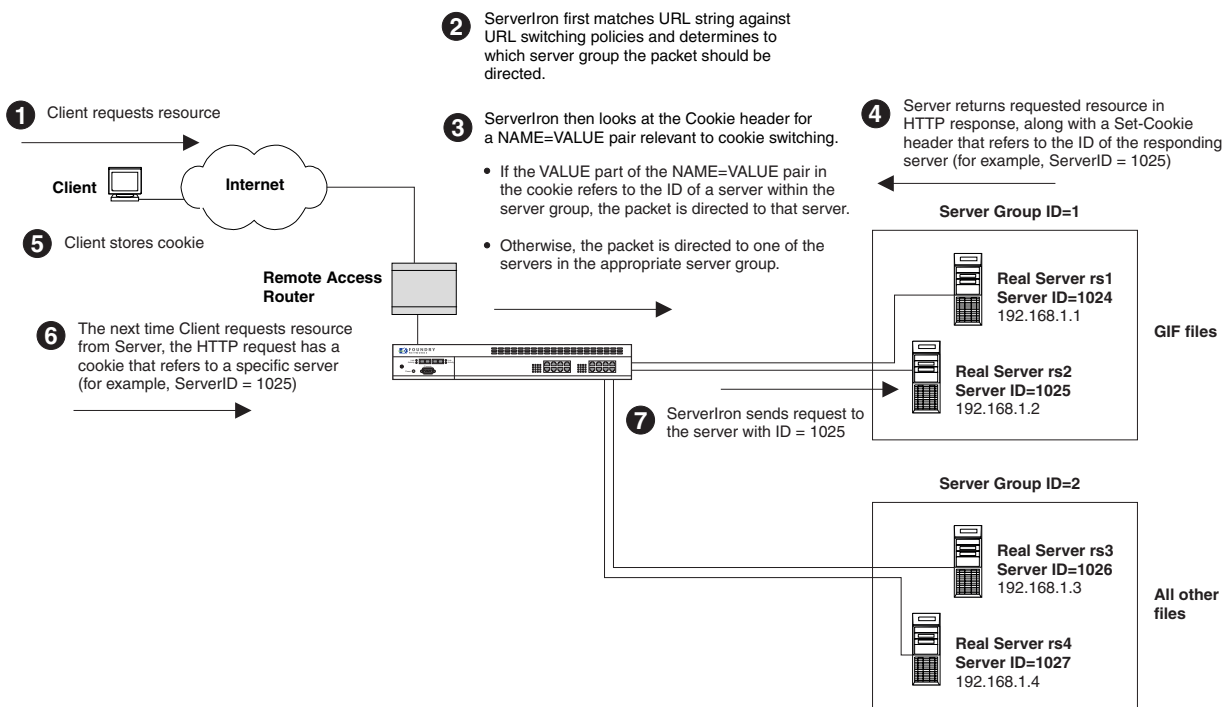


## Using URL Switching and Cookie Switching Concurrently

You can configure the ServerIron to use URL switching and cookie switching at the same time. When URL switching and cookie switching are enabled concurrently, the ServerIron directs HTTP request messages to real servers based first on the contents of the URL string and then on the contents of the Cookie header (if any).

The configuration in Figure 11.5 illustrates how the ServerIron directs HTTP requests when URL switching and cookie switching are used concurrently.

**Figure 11.5 Concurrent URL and cookie switching**



This is what's going on in the diagram:

1. The client requests a resource (for example /images/foundry.gif) from the server.
2. The ServerIron first matches the URL string against URL switching policies and determines to which server group the HTTP request should be directed.
3. After selecting a server group for the request, the ServerIron looks at the Cookie header for a NAME=VALUE pair relevant to cookie switching (for example, ServerID=1025).
  - If such a NAME=VALUE pair is found, the ServerIron directs the request to the specific real server whose ID corresponds to the VALUE part of the NAME=VALUE pair.
  - If no NAME=VALUE pair is found, the request is directed to one of the load-balanced real servers in the server group determined in Step 2.
4. When the server responds to the request, the HTTP response message contains a Set-Cookie header that includes a NAME=VALUE pair relevant to cookie switching. The NAME part is a user-defined token (for example, ServerID) that identifies the cookie; the VALUE part refers to the ID of the specific real server that responded to the request.
5. The NAME=VALUE pair in the Set-Cookie header is stored on the client.
6. The next time the client requests a resource from the server, the HTTP request contains a Cookie header that includes the NAME=VALUE pair.

7. Using the procedure in Steps 2 – 3, the ServerIron directs the HTTP request to the real server whose ID corresponds to the VALUE part of the NAME=VALUE pair.

The illustration above shows a configuration where the GIF images are stored on servers in server group ID = 1, and all other files are stored on servers in server group ID = 2. When a client sends an HTTP request that contains the URL string “/images/foundry.gif”, the request would be handled as follows:

- The first time the client requests this resource, the ServerIron uses a URL switching policy to determine the server to which to direct the request. The URL switching policy has selection criteria that specifies HTTP requests containing URL strings ending with “gif” are to be sent to one of the load-balanced real servers in server group ID = 1
- Since the HTTP request has no NAME=VALUE pair relevant to Cookie switching, the ServerIron forwards the request to one of the servers in the server group (rather than a specific real server); for example, real server rs2.
- The HTTP response sent back by the real server includes a Set-Cookie header that contains a NAME=VALUE pair that refers to the server ID. In the configuration above, real server rs2 would send back a Set-Cookie header with a NAME=VALUE pair of ServerID=1025. This NAME=VALUE pair is stored on the client.
- The next time the client requests this resource, the Cookie header in the HTTP request contains this NAME=VALUE pair. The ServerIron uses a URL switching policy to determine the server to which to direct the request, then directs the HTTP request to the server indicated by the VALUE part of the NAME=VALUE pair. In the configuration above, an HTTP request that has a Cookie header containing ServerID=1025 would be directed to the real server with the ID of 1025; that is, real server rs2.

Setting up concurrent URL and cookie switching consists of the following steps:

1. Setting up the URL switching policies
2. Configuring server groups and server IDs
3. Configuring the servers to send a Set-Cookie header that contains a NAME=VALUE pair relevant to cookie switching
4. Setting up the virtual server

## Setting Up URL Switching Policies

The procedure for setting up URL switching policies for concurrent URL and cookie switching is the same as the one described in “Setting Up URL Switching Policies” on page 11-3.

For example, the following URL switching policy directs HTTP requests containing URL strings that end with “gif” to server group ID = 1 and directs all other HTTP requests to server group ID = 2.

```
ServerIron(config)# url-map gifPolicy
ServerIron(config-url-gifPolicy)# method suffix
ServerIron(config-url-gifPolicy)# match "gif" 1
ServerIron(config-url-gifPolicy)# default 2
ServerIron(config-url-gifPolicy)# exit
```

## Configuring Server Groups and Server IDs

When you configure concurrent URL and cookie switching, you place each server in a server group and assign it a server ID. The server group is used for URL switching, and the server ID is used for cookie switching.

For example, the following commands place real server rs2 in Figure 11.5 in server group ID = 1 and assign real server rs2 a server ID of 1025.

```
ServerIron(config)# server real-name rs2 192.168.1.2
ServerIron(config-rs-rs2)# port http group-id 1 1
ServerIron(config-rs-rs2)# port http server-id 1025
ServerIron(config-rs-rs2)# exit
```

**Syntax:** port http group-id <server-group-id-pairs>

**Syntax:** port http server-id <server-id>

The **port http group-id** command indicates the server group to which the real server belongs. The server group is expressed as a pair of numbers, indicating a range of real server group IDs. The first number is the lowest-numbered server group ID, and the second is the highest-numbered server group ID. In this example, the real server belongs only to the server group with ID = 1, so the last two numbers in the **port http group-id** command are 1 1 (note the space between the two numbers). Valid numbers for server group IDs are 0 – 1023. See “Configuring the Real Servers” on page 11-7 for more information on setting up server groups.

The **port http server-id** command sets the server ID for the real server at 1025. HTTP requests coming into the ServerIron that have a cookie value that refers to this server ID are always sent to this real server. Valid numbers for server IDs are 1024 – 2047.

## Configuring the Server to Set a Cookie

In concurrent URL and cookie switching, you configure your server to include a Set-Cookie header in its responses to HTTP requests. This Set-Cookie header must include a NAME=VALUE pair that refers to the ID of the server.

**NAME** Is a token that identifies the cookie. This can be any word (for example, ServerID), but cannot start with a dollar sign (\$).

**VALUE** Refers to the ID of a real server (1024 – 2047). See “Configuring Server Groups and Server IDs” on page 11-28 for information on setting up IDs for real servers.

The exact procedure for configuring a server to send a Set-Cookie header depends on your server configuration and is not discussed here. However, the following is an example of a JavaScript command to create a cookie whose NAME is ServerID and VALUE is 1025:

```
SetCookie("ServerID","1025")
```

Consult RFC 2109 or your JavaScript documentation for more information on the Set-Cookie header.

## Enabling Concurrent URL and Cookie Switching on the Virtual Server

The following commands enable concurrent URL and cookie switching on the virtual server in Figure 11.5 on page 11-27:

```
ServerIron(config)# server virtual URLcookieVIP 192.168.1.242
ServerIron(config-vs-URLcookieVIP)# port http
ServerIron(config-vs-URLcookieVIP)# port http url-map gifPolicy
ServerIron(config-vs-URLcookieVIP)# port http cookie-name ServerID
ServerIron(config-vs-URLcookieVIP)# port http url-cookie-switching
ServerIron(config-vs-URLcookieVIP)# bind http rs1 http rs2 http rs3 http rs4 http
ServerIron(config-vs-URLcookieVIP)# exit
```

**Syntax:** port http url-map <policy-name>

**Syntax:** port http cookie-name <name>

**Syntax:** port http url-cookie-switching

The **port http url-map** command specifies the URL switching policy to be active for this VIP.

The **port http cookie-name** command specifies the name of the cookie to be used in cookie switching. This must be the same as the NAME token you configured your server to include in responses to HTTP requests. In the example, the cookie name is “ServerID”.

The **port http url-cookie-switching** command enables concurrent URL and cookie switching on this virtual server. In this example, when the ServerIron encounters an HTTP request sent to this VIP, the URL string in the request is matched against the selection criteria in URL switching policy “gifPolicy”, and the ServerIron determines to which server group the request should be sent. The ServerIron then looks in the Cookie header for a cookie with ServerID as the NAME part of the NAME=VALUE pair. If the cookie is found, the request is directed to the real server whose ID is specified in the VALUE part of the NAME=VALUE pair. If the cookie is not found, the request is directed to the server group determined by the URL switching policy.

## Configuring HTTP Header Hashing

HTTP header hashing is another way the ServerIron can make forwarding decisions based on the contents of an HTTP request message. In HTTP header hashing, the ServerIron examines information in the HTTP request (either the Cookie header or the URL string) and internally maps this information to one of the real servers bound to the virtual server. This HTTP request and all future HTTP requests that contain this information then always go to the same real server.

For example, an HTTP request might have a URL string that consists of the text “/download/files/myfile.html”. The ServerIron would internally map this URL string to a real server bound to the virtual server. The next time an HTTP request that has this exact same URL string comes into the VIP, it would go to the same real server as the first one did.

Unlike URL or cookie switching, HTTP header hashing directs HTTP requests to a specific real server, rather than to a server group. In addition, since the mapping process takes place internally and automatically, you do not create switching policies or configure your server to send a cookie to the client.

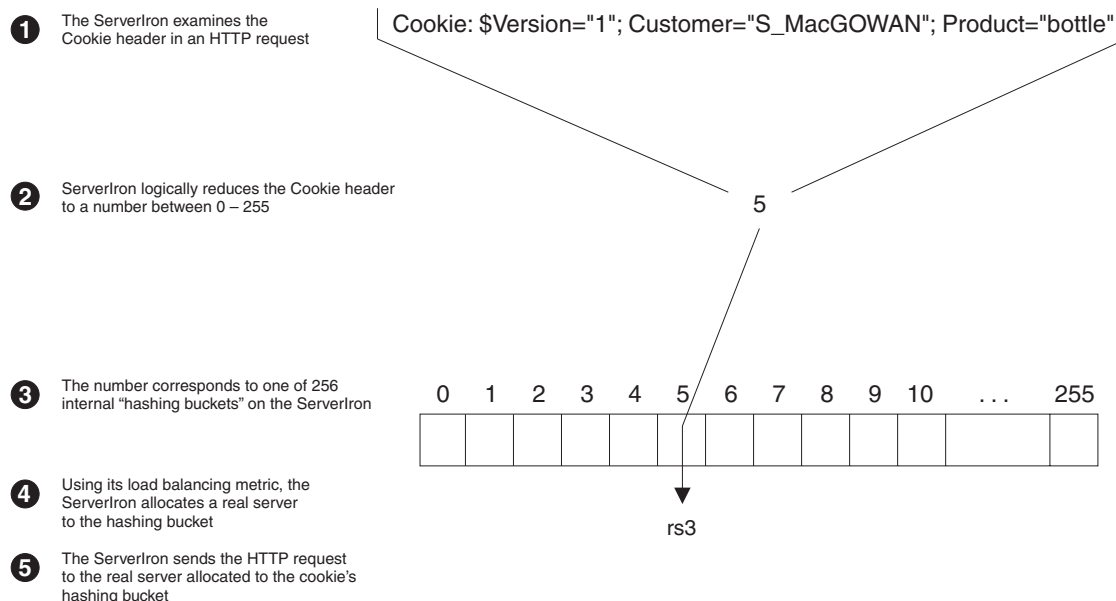
The ServerIron can perform four kinds of HTTP header hashing:

- **Cookie hashing** causes HTTP requests that contain the same Cookie header always to go to the same real server.
- **Selective cookie hashing** looks for user-specified cookies in the Cookie header. HTTP requests that contain the same values in the user-specified cookies always go to the same real server.
- **URL string hashing** causes HTTP requests that contain the same URL string always to go to the same real server.
- **URL segment hashing** maps a segment of a URL string to a server group. HTTP requests that contain this same URL segment always go to the same server group. Unlike Cookie hashing or URL string hashing, URL segment hashing sends HTTP requests to a load-balanced server group, rather than to an actual real server.

### Cookie Hashing

Cookie hashing causes HTTP requests that contain the same Cookie header always to go to the same real server. When an HTTP request comes into a virtual server, the ServerIron examines its Cookie header and automatically selects a real server from among those bound to the virtual server. The HTTP request, as well as all subsequent HTTP requests that contain the same Cookie header, go to that real server.

Figure 11.6 illustrates how the ServerIron uses cookie hashing to direct HTTP requests to a real server.

**Figure 11.6 Using cookie hashing to select a real server**

This is what's going on in the diagram:

1. The ServerIron examines the Cookie header in an HTTP request sent to the virtual server.
2. The ServerIron assigns a number between 0 – 255 to the contents of the Cookie header.
3. This number corresponds to a hashing bucket on the ServerIron.
4. Using its load balancing metric, the ServerIron allocates one of the real servers bound to the virtual server to the hashing bucket. Possible load balancing metrics are least connections, weighted percentage, and round robin. By default, the least connections metric is applied globally to all virtual servers. If you define a metric specifically for this virtual server, that metric takes precedence over the globally defined metric.
5. The ServerIron directs the HTTP request to the real server assigned to the cookie's hashing bucket. All future HTTP requests that have the same Cookie header are sent to the same real server.

This means that in the example in Figure 11.6 on page 11-31, HTTP requests that have a Cookie header consisting of the text "Cookie: Version="1"; Customer="S\_MacGOWAN"; Product="bottle"" are always sent to real server rs3.

### USING THE CLI

The following commands enable cookie hashing on a virtual server:

```
ServerIron(config)# server virtual cookieHash 209.157.22.241
ServerIron(config-vs-cookieHash)# port http
ServerIron(config-vs-cookieHash)# port http cookie-hashing
ServerIron(config-vs-cookieHash)# bind http rs1 http
ServerIron(config-vs-cookieHash)# bind http rs2 http
ServerIron(config-vs-cookieHash)# bind http rs3 http
ServerIron(config-vs-cookieHash)# exit
```

**Syntax:** port http cookie-hashing

The **port http cookie-hashing** command enables cookie hashing on the virtual server. Note that you cannot have cookie hashing and cookie switching enabled on the same virtual server.

The **bind http** commands bind the real servers to the VIP. The ServerIron allocates these real servers to its hashing buckets.

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of server load balancing option links.
4. Select the Virtual Server link to display the Virtual Server panel.
5. In the Server Name field, enter the name of the virtual server.
6. In the Server IP field, enter the IP address of the virtual server.
7. Click the Add button to add the virtual server to the device's running-config file.
8. Click the Virtual Server Port link at the bottom of the panel to display the Virtual Server Port panel.
9. Click the Modify button for the virtual server you just created.
10. Click the checkbox next to Cookie Hashing.
11. Click the Bind link at the bottom of the panel. The Bind panel is displayed.
12. Select the virtual server from the Virtual Server Name pulldown menu.
13. Select HTTP from the Virtual TCP/UDP Port pulldown menu.
14. From the Real Server Name pulldown menu, select a real server that you are binding to this virtual server.
15. Click Bind to bind the real server to the virtual server.
16. Repeat Steps 14 – 15 for each real server you want to bind to this virtual server.
17. Select the Save link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Clearing Cookie Hashing Bucket Allocations and Statistics

In a cookie hashing configuration, you can reset the hashing bucket allocations and clear the statistics about the number of hits each bucket has received.

To reset the cookie hashing bucket allocations, enter commands such as the following:

```
ServerIron(config)# server virtual cookieHash 209.157.22.241
ServerIron(config-vs-cookieHash)# port http clear-hash-buckets
ServerIron(config-vs-cookieHash)# exit
```

**Syntax:** port http clear-hash-buckets

To reset the cookie hashing bucket statistics, enter commands such as the following:

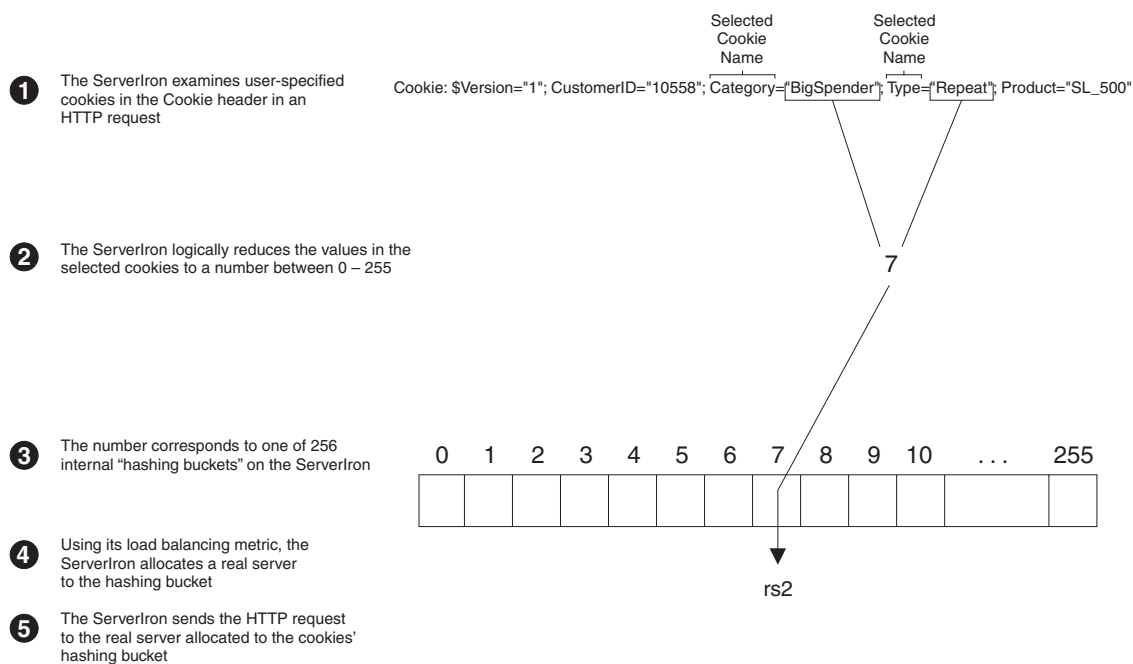
```
ServerIron(config)# server virtual cookieHash 209.157.22.241
ServerIron(config-vs-cookieHash)# port http clear-hash-statistics
ServerIron(config-vs-cookieHash)# exit
```

**Syntax:** port http clear-hash-statistics

### Selective Cookie Hashing

Selective cookie hashing selects a real server based on values in user-specified cookies, rather than the entire Cookie header. HTTP requests that contain the same values in the user-specified cookies always go to the same real server. When an HTTP request comes into a virtual server, the ServerIron looks for user-specified cookie names in the Cookie header and maps their values to one of the real servers bound to the virtual server. The HTTP request, as well as all subsequent HTTP requests that contain the same values in the user-specified cookies, go to that real server.

Figure 11.7 illustrates how the ServerIron uses selective cookie hashing to direct HTTP requests to a real server.

**Figure 11.7 Using selective cookie hashing to select a real server**

This is what's going on in the diagram:

1. The ServerIron searches the Cookie header in an HTTP request for user-specified cookie names (that is, the NAME part of the NAME=VALUE pair). You can specify up to five cookie names. In the example above, the user-specified cookie names are "Category" and "Type".
2. If one or more of the user-specified cookie names are found in the Cookie header, the ServerIron assigns a number between 0 – 255 to the values (that is, the VALUE part of the NAME=VALUE pair) of the selected cookies.
3. This number corresponds to a hashing bucket on the ServerIron.
4. Using its load balancing metric, the ServerIron allocates one of the real servers bound to the virtual server to the hashing bucket. Possible load balancing metrics are least connections, weighted percentage, and round robin. By default, the least connections metric is applied globally to all virtual servers. If you define a metric specifically for this virtual server, that metric takes precedence over the globally defined metric.
5. The ServerIron directs the HTTP request to the real server assigned to the cookie values' hashing bucket. All future HTTP requests that have the user-specified cookies containing the same values are sent to the same real server.

This means that in the example in Figure 11.7, HTTP requests that have a Cookie header with NAME=VALUE pairs "Category="BigSpender"" and "Type="Repeat"" are always sent to real server `rs2`. If none of the selected cookies are found in the Cookie header, standard cookie hashing takes place, as described in "Cookie Hashing" on page 11-30.

#### USING THE CLI

The following commands enable selective cookie hashing on a virtual server:

```
ServerIron(config)# server virtual cookieHash 192.168.1.123
ServerIron(config-vs-cookieHash)# port http
ServerIron(config-vs-cookieHash)# port http cookie-hash-name Category
ServerIron(config-vs-cookieHash)# port http cookie-hash-name Type
ServerIron(config-vs-cookieHash)# port http cookie-hashing
ServerIron(config-vs-cookieHash)# bind http rs1 http
ServerIron(config-vs-cookieHash)# bind http rs2 http
```

```
ServerIron(config-vs-cookieHash)# bind http rs3 http
ServerIron(config-vs-cookieHash)# exit
```

**Syntax:** port http cookie-hash-name <cookie-name>

The **port http cookie-hash-name** commands specify the names of the cookies to be used in selective cookie hashing. This is the NAME part of a cookie's NAME=VALUE pair. You can specify up to five cookie names.

If one or more of the selected cookie names are found in a Cookie header, the ServerIron performs hashing based on their values. In the sample configuration above, one of the following can take place:

- If a Cookie header has both of the selected cookies, hashing is performed based on the values in the two cookies. All requests that contain the same two NAME=VALUE pairs are always sent to the same real server.
- If a Cookie header has only one of the selected cookies (for example "Category", but not "Type"), hashing is performed based on the value in the single cookie. All requests that contain the same NAME=VALUE pair are always sent to the same real server.
- If neither of the selected cookies are found in the Cookie header, standard cookie hashing takes place, as described in "Cookie Hashing" on page 11-30. Hashing is performed based on the entire Cookie header.

#### [USING THE WEB MANAGEMENT INTERFACE](#)

---

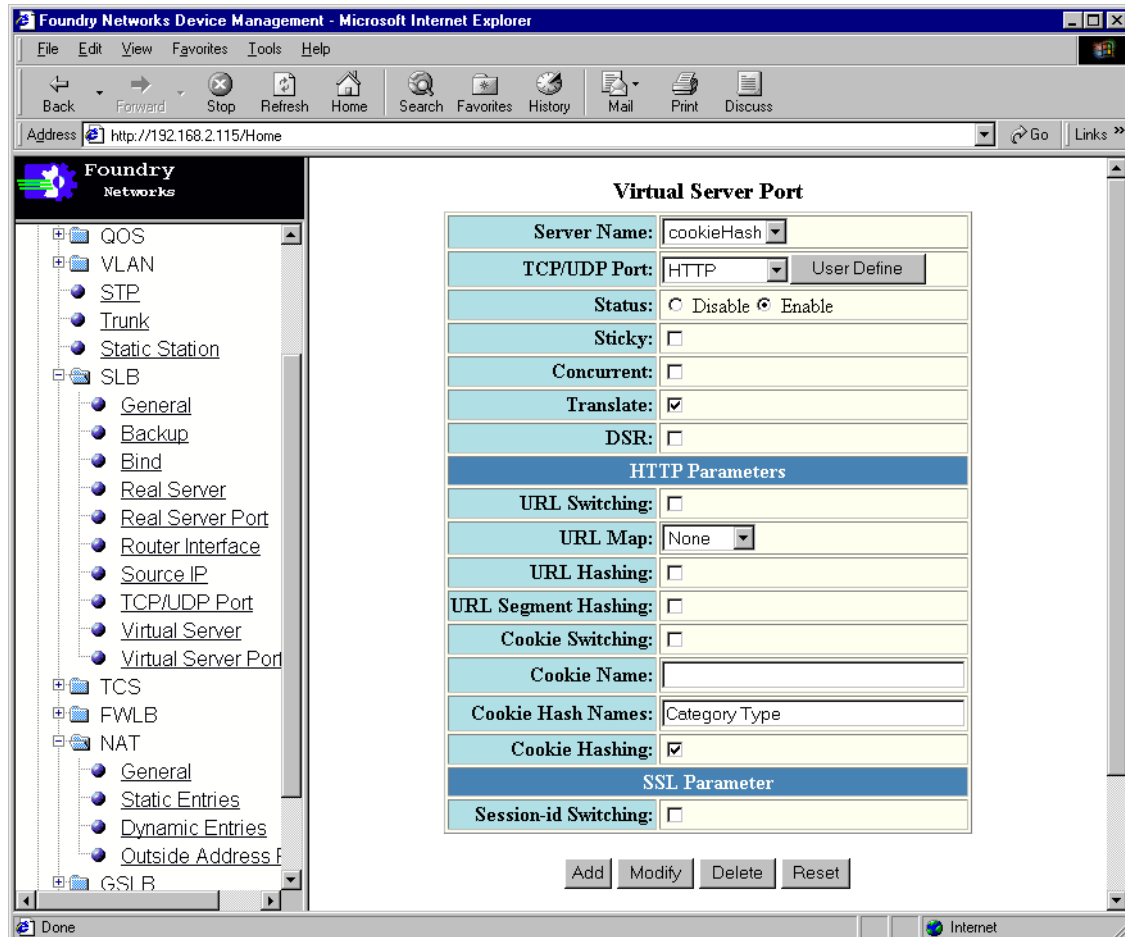
**NOTE:** The following procedure applies to the ServerIron 400 and ServerIron 800 only. For other versions of the ServerIron, use the CLI to configure this feature.

---

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of server load balancing option links.
4. Select the [Virtual Server](#) link to display the Virtual Server panel.
5. In the Server Name field, enter the name of the virtual server.
6. In the Server IP field, enter the IP address of the virtual server.
7. Click the Add button to add the virtual server to the device's running-config file.
8. Click the [Virtual Server Port](#) link at the bottom of the panel to display the Virtual Server Port panel.
9. Click the Modify button for the virtual server you just created.



The following panel is displayed:



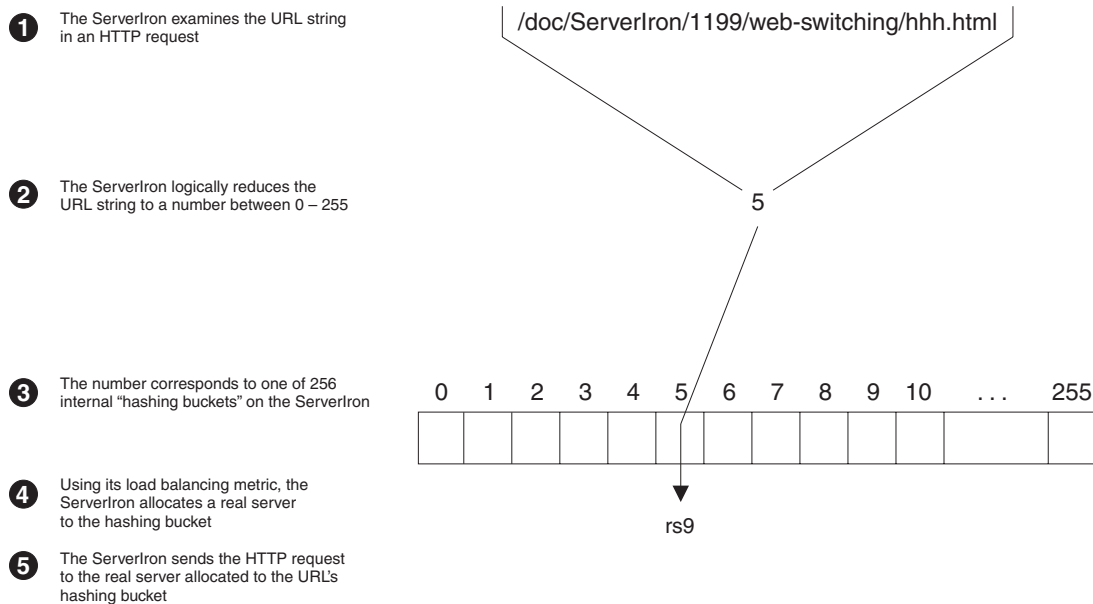
1. Click the checkbox next to Cookie Hashing.
2. In the Cookie Hash Names field, enter the names of the cookies to be used in selective cookie hashing. This is the NAME part of a cookie's NAME=VALUE pair. You can specify up to five cookie names.
3. Click the Bind link at the bottom of the panel. The Bind panel is displayed.
4. Select the virtual server from the Virtual Server Name pulldown menu.
5. Select HTTP from the Virtual TCP/UDP Port pulldown menu.
6. From the Real Server Name pulldown menu, select a real server that you are binding to this virtual server.
7. Click Bind to bind the real server to the virtual server.
8. Repeat Steps 6 – 7 for each real server you want to bind to this virtual server.
9. Select the Save link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## URL String Hashing

URL string hashing works similarly to cookie hashing. The only difference is that in URL string hashing, the URL string, not the cookie header, is reduced to a number that corresponds to one of the ServerIron's hashing buckets.

Figure 11.8 illustrates how the ServerIron uses URL string hashing to direct HTTP requests to a real server.

**Figure 11.8 Using URL string hashing to select a real server**



This is what's going on in the diagram:

1. The ServerIron examines the URL string in an HTTP request sent to the VIP.
2. The ServerIron assigns a number between 0 – 255 to the contents of the URL string.
3. This number corresponds to a hashing bucket on the ServerIron.
4. Using its load balancing metric, the ServerIron allocates one of the real servers bound to the virtual server to the hashing bucket. Possible load balancing metrics are least connections, weighted percentage, and round robin. By default, the least connections metric is applied globally to all virtual servers. If you define a metric specifically for this virtual server, that metric takes precedence over the globally defined metric.
5. The ServerIron directs the HTTP request to the real server matching the URL's hashing bucket. All future HTTP requests that have the same URL string are sent to the same real server.

In the example in Figure 11.8, HTTP requests that have a URL string consisting of the text `/doc/1199/web-switching/hhh.html` are always sent to real server `rs9`.

### USING THE CLI

The following commands enable URL string hashing on a virtual server:

```
ServerIron(config)# server virtual URLstringHash 209.157.22.241
ServerIron(config-vs-URLstringHash)# port http
ServerIron(config-vs-URLstringHash)# port http hash-url-string
ServerIron(config-vs-URLstringHash)# bind http rs7 http
ServerIron(config-vs-URLstringHash)# bind http rs8 http
ServerIron(config-vs-URLstringHash)# bind http rs9 http
ServerIron(config-vs-URLstringHash)# exit
```

**Syntax:** port http hash-url-string

The **port http hash-url-string** command enables URL string hashing on the virtual server.

The **bind http** commands bind the real servers to the VIP. The ServerIron allocates these real servers to its hashing buckets.

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.

2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of server load balancing option links.
4. Select the Virtual Server link to display the Virtual Server panel.
5. In the Server Name field, enter the name of the virtual server.
6. In the Server IP field, enter the IP address of the virtual server.
7. Click the Add button to add the virtual server to the device's running-config file.
8. Click the Virtual Server Port link at the bottom of the panel to display the Virtual Server Port panel.
9. Click the Modify button for the virtual server you just created.
10. Click the checkbox next to URL Hashing.
11. Click the Bind link at the bottom of the panel. The Bind panel is displayed.
12. Select the virtual server from the Virtual Server Name pulldown menu.
13. Select HTTP from the Virtual TCP/UDP Port pulldown menu.
14. From the Real Server Name pulldown menu, select a real server that you are binding to this virtual server.
15. Click Bind to bind the real server to the virtual server.
16. Repeat Steps 14 – 15 for each real server you want to bind to this virtual server.
17. Select the Save link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

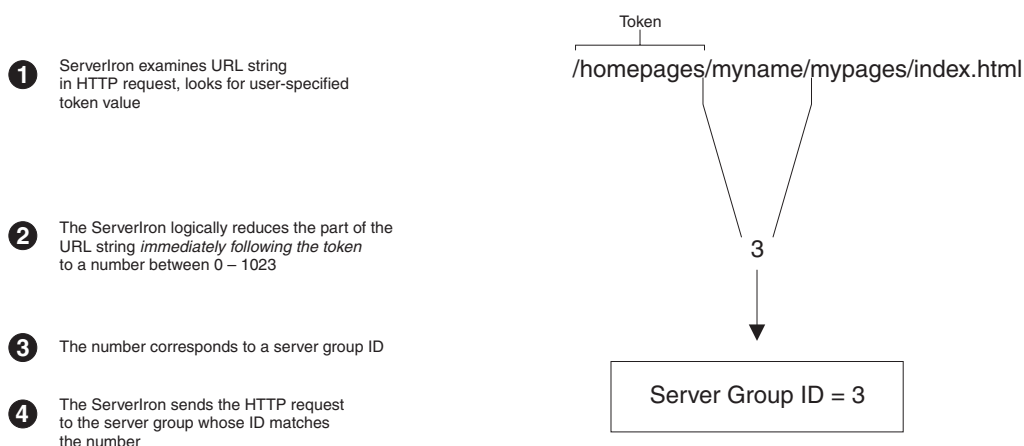
## URL Segment Hashing

In URL segment hashing, the ServerIron searches the URL string in an HTTP request for a user-defined token and uses this token to map HTTP requests to a server group.

- If the user-defined token is found in the URL string, the ServerIron uses the segment of the URL string ***immediately following*** the token to map HTTP requests to a server group.
- If the token is not found in the URL string, the ServerIron uses the segment at the beginning of the URL string to map HTTP requests to a server group.

Figure 11.9 illustrates how the ServerIron selects a server group when the user-defined token is part of the URL string.

**Figure 11.9 Using URL segment hashing to direct HTTP requests to a server group (token found)**



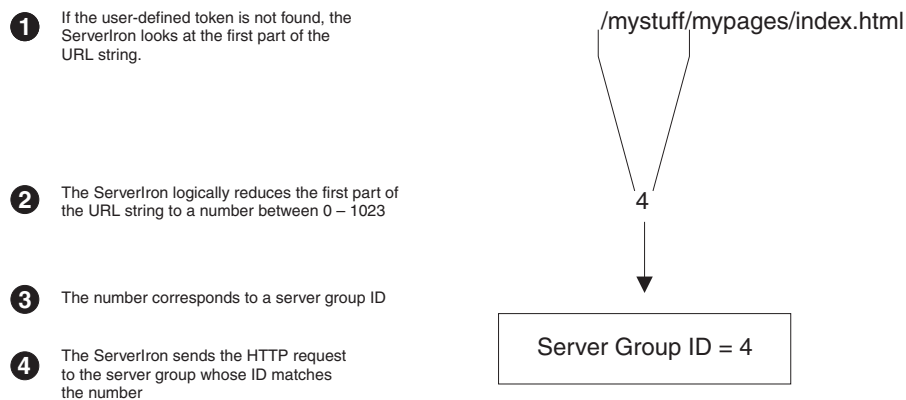
This is what's going on in the diagram:

1. The ServerIron searches the URL string in an HTTP request for a user-defined token. In the example above, the token is “homepages”. You can define multiple tokens that apply to this VIP.
2. The ServerIron assigns a number between 0 – 1023 to the segment of the URL string that immediately follows the token. In the example above, the segment of the URL string that immediately follows the token is “myname”.
3. This number corresponds to the ID of a server group.
4. The ServerIron directs the HTTP request to the server group whose ID matches this number. All future HTTP requests that have a URL string *containing the token followed by the segment* are sent to the same server group.

This means that in the example in Figure 11.9, HTTP requests that have “/homepages/myname” anywhere within the URL string are always sent to one of the load-balanced real servers in server group ID = 3.

If the user-defined token(s) are not found in the URL string, the ServerIron uses the first part of the URL string to map HTTP requests to a real server. Figure 11.10 illustrates how this works.

**Figure 11.10 Using URL segment hashing to direct HTTP requests to a server group (token(s) not found)**



When none of the user-defined tokens appear in the URL string, the ServerIron uses the first part of the URL string to map HTTP requests to a server group. In the example above, the first part of the URL string is “mystuff”. All HTTP requests that have a URL string that begins with the segment “mystuff” would be sent to one of the load-balanced real servers in server group ID = 4.

### Setting Up the Server Groups

See “Configuring the Real Servers” on page 11-7 for information on setting up real servers and assigning them to server groups.

### Setting up the Virtual Server

To configure the virtual server for URL segment hashing, use one of the following methods.

#### USING THE CLI

The following commands enable URL segment hashing on a virtual server:

```
ServerIron(config)# server virtual URLsegmentHash 209.157.22.241
ServerIron(config-vs-URLsegmentHash)# port http
ServerIron(config-vs-URLsegmentHash)# port http hash-segment homepages
ServerIron(config-vs-URLsegmentHash)# port http hash-segment awaypages
ServerIron(config-vs-URLsegmentHash)# port http hash-screen-name
ServerIron(config-vs-URLsegmentHash)# bind http rs1 http
ServerIron(config-vs-URLsegmentHash)# bind http rs2 http
ServerIron(config-vs-URLsegmentHash)# exit
```

**Syntax:** port http hash-segment <token>

**Syntax:** port http hash-screen-name

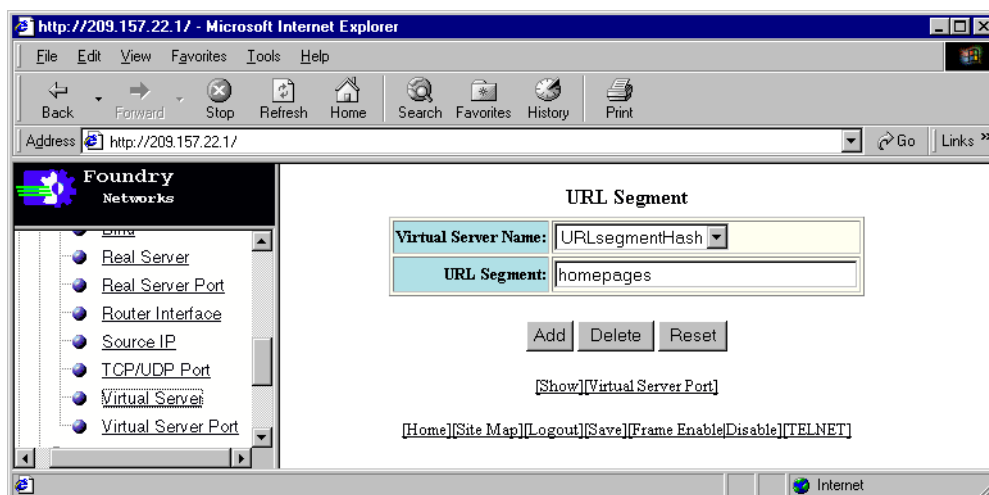
The **port http hash-segment** command specifies the token to be used for URL segment hashing. Note that you can have multiple **port http hash-segment** commands in a virtual server configuration.

The **port http hash-screen-name** command enables URL segment hashing on the virtual server

The **bind http** commands bind the real servers to the VIP.

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of server load balancing option links.
4. Select the Virtual Server link to display the Virtual Server panel.
5. In the Server Name field, enter the name of the virtual server.
6. In the Server IP field, enter the IP address of the virtual server.
7. Click the Add button to add the virtual server to the device's running-config file.
8. Click the Virtual Server Port link at the bottom of the panel to display the Virtual Server Port panel.
9. Click the Modify button for the virtual server you just created.
10. Click the checkbox next to URL Segment Hashing.
11. Click the URL Segment link at the bottom of the panel. The following panel is displayed:



12. Select the virtual server from the Virtual Server Name pulldown menu.
13. In the URL Segment field, enter the token to be used for URL segment hashing.
14. Click the Add button to save the changes to the device's running-config file.
15. Click the Virtual Server Port link at the bottom of the panel to display the Virtual Server Port panel.
16. Click the Modify button for the virtual server you just created.
17. Click the Bind link at the bottom of the panel. The Bind panel is displayed.
18. Select the virtual server from the Virtual Server Name pulldown menu.
19. Select HTTP from the Virtual TCP/UDP Port pulldown menu.
20. From the Real Server Name pulldown menu, select a real server that you are binding to this virtual server.

21. Click Bind to bind the real server to the virtual server.
22. Repeat Steps 20 – 21 for each real server you want to bind to this virtual server.
23. Select the [Save](#) link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Displaying Hashing Bucket Statistics

You can display information about hashing bucket assignments and the number of hits each bucket has received. Enter the following command:

```
ServerIron# show server hash
```

**Syntax:** show server hash

## Configuring SSL Session ID Switching

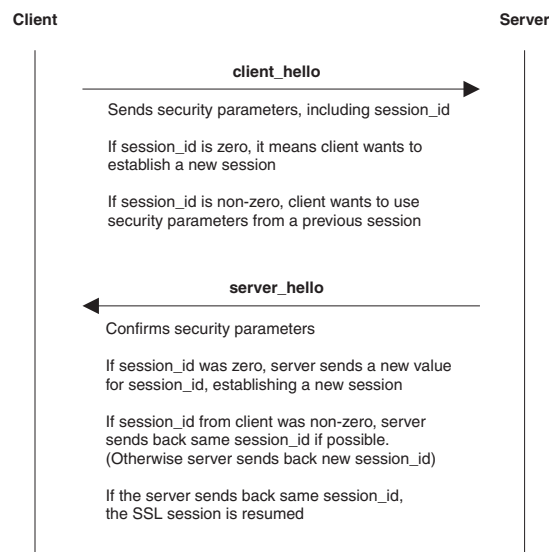
SSL Session ID switching is the ServerIron's ability to connect a client to the same real server to which it had previously established an SSL (Secure Sockets Layer) connection.

SSL is used to provide security in web transactions. An SSL connection is initiated when a user clicks a hyperlink that begins with "https" (for example, <https://secure.foundrynet.com>). The browser (client) initiates an SSL connection with the server on TCP port 443, a secure link is negotiated, and encrypted data is transferred across it.

The SSL Handshake Protocol (SSLHP), one of two component protocols of SSL, negotiates the connection between the client and server. SSLHP establishes security parameters for an SSL session, including the SSL version number and the method of data encryption to use. One of the security parameters set by SSLHP is the **SSL Session ID**, a variable length value contained in the session\_id field in SSLHP messages. The SSL Session ID indicates whether the client wants to use the security parameters established in a previous session or establish a completely new connection.

Figure 11.11 illustrates how the initial SSLHP messages exchanged between a client and server, client\_hello and server\_hello, establish an SSL Session ID.

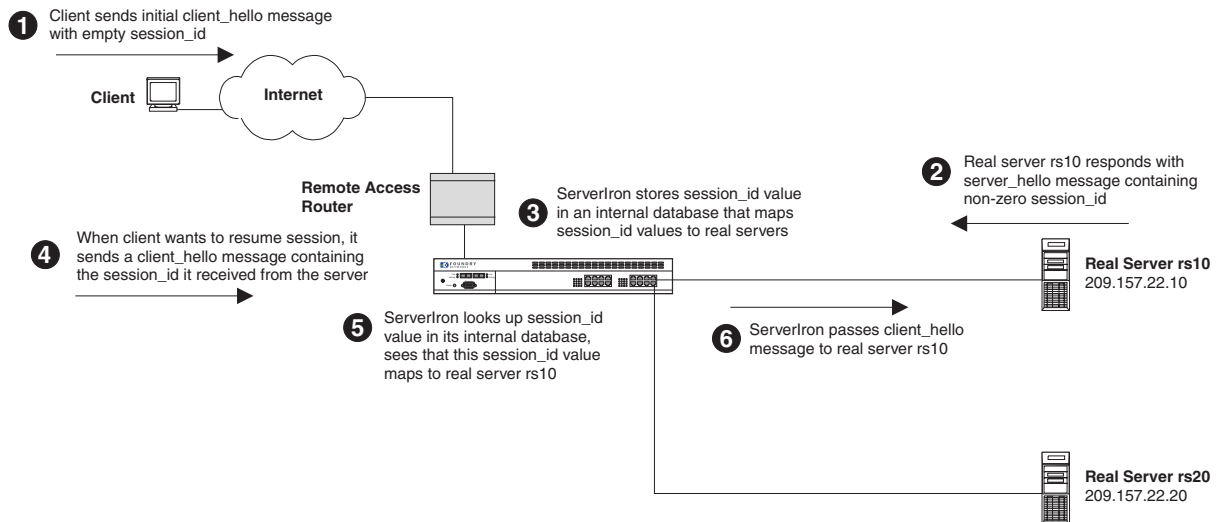
**Figure 11.11 How the SSL Handshake Protocol establishes a Session ID**



If the value in the session\_id field that the client sends to the server is non-zero, the ServerIron can connect the client to the server that originally sent the Session ID value. Figure 11.12 illustrates how this function, called **SSL Session ID switching**, works.

**NOTE:** SSL Session ID switching is supported for SSL v3.0 and higher only. In SSL versions prior to 3.0, the session ID was established later in the handshaking process, after the client and server had started exchanging encrypted data. If the session ID is encrypted, the ServerIron cannot make forwarding decisions based on this information.

**Figure 11.12 How the ServerIron uses an SSL Session ID to select a real server**



This is what's going on in the diagram:

1. The first time a client attempts to establish an SSL connection to the server, there is no history of a previous SSL session, so the session\_id field in the client\_hello message it sends to the server is empty.
2. The server (in this example, real server rs10) sees that the session\_id field in the client\_hello message is empty, indicating the client wants to establish a new SSL session. The server responds to the client with a server\_hello message that contains a session\_id field with a non-zero value.
3. The ServerIron examines the value in the session\_id field sent by the server. The ServerIron adds this value to an internal database, associating it with the real server that sent it. This association between the session\_id value and the real server resides in the ServerIron's database for a user-specified amount of time (default 30 minutes), after which it is aged out. In this example, the ServerIron would map the value in the session\_id field to real server rs10.
4. When the client resumes the SSL connection to the server, it sends a client\_hello message containing the session\_id value sent by the server.
5. The ServerIron examines the value in the session\_id field sent by the client and looks it up in its internal database.
6. If the value in the session\_id field maps to a real server, the ServerIron initiates a TCP connection to the server and passes the client\_hello message to it. The ServerIron forwards subsequent packets between the client and server with modifications to the IP and TCP header for sequence number, acknowledgment number, and checksum adjustment.

Setting up SSL session ID switching consists of the following steps:

1. Configuring the real servers for SSL
2. Configuring the virtual server for SSL session ID switching
3. Adjusting the age timer in the ServerIron's database (optional)
4. Adjusting the maximum number of session\_id-to-real-server associations the ServerIron can store in its internal database (optional)

These tasks are described in the following sections.

## Configuring Real Servers for SSL

To configure the real servers shown in Figure 11.12 for SSL, use one of the following methods.

### *USING THE CLI*

To configure the real servers for SSL:

```
ServerIron(config)# server real-name rs10 207.157.22.10
ServerIron(config-rs-rs10)# port ssl
ServerIron(config-rs-rs10)# exit

ServerIron(config)# server real-name rs20 207.157.22.20
ServerIron(config-rs-rs20)# port ssl
ServerIron(config-rs-rs20)# exit
```

**Syntax:** server real-name <real-server-name> <ip-addr>

**Syntax:** port ssl

The **server real-name** commands define the names and IP addresses of the real servers.

The **port ssl** commands add port 443 (SSL) to the real servers.

### *USING THE WEB MANAGEMENT INTERFACE*

To configure a real server for SSL, use the following procedure. Repeat this procedure for each real server to be used in SSL Session ID switching.

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of server load balancing option links.
4. Select the Real Server link. The Real Server panel is displayed.
5. In the Server Name field, enter the name of the real server.
6. In the Server IP field, enter the IP address of the real server.
7. Click the Add button to add the real server to the device's running-config file.
8. Click the Real Server Port link at the bottom of the panel. The Real Server Port panel is displayed.



9. Click the Modify button for the real server you just created. The following panel is displayed:

**Real Server Port**

Server Name: rs10

TCP/UDP Port: SSL User Define

Status: ☐ Disable ☒ Enable

Keep Alive: ☐

**DNS Parameters**

+DNS Zone:

+Addr Query:

+Proxy: ☐

**HTTP Parameters**

\*Method: HEAD

\*URL:

\*Status Code:

**Group Id Range**

From	To

Add Modify Delete Reset

[Show Real Server Port]

\* -> HTTP Only, +> DNS Only

[Virtual Server][Virtual Server Port][Real Server][Bind]

[Home][Site Map][Logout][Save][Frame Enable/Disable][TELNET]

10. Select the real server from the Server Name pulldown menu.
11. Select SSL from the TCP/UDP Port pulldown menu.
12. Click the Add button to save the changes to the device's running-config file.
13. Select the Save link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Configuring the Virtual Server for SSL Session ID Switching

To configure the virtual server for SSL Session ID switching, use one of the following methods.

**NOTE:** You cannot use an access control list on a VIP that has SSL Session ID switching enabled.

### USING THE CLI

The following commands enable SSL Session ID switching on a virtual server called sslVIP:

```
ServerIron(config)# server virtual sslVIP 209.157.22.241
ServerIron(config-vs-sslVIP)# port ssl session-id-switching
ServerIron(config-vs-sslVIP)# bind ssl rs10 ssl
ServerIron(config-vs-sslVIP)# bind ssl rs20 ssl
ServerIron(config-vs-sslVIP)# exit
```

**Syntax:** port ssl session-id-switching

**Syntax:** port <port-number> session-id-switching

**Syntax:** bind ssl <real-server-name> ssl

The **port ssl session-id-switching** command enables SSL Session ID switching on this virtual server.

The **bind ssl** commands bind the virtual server to SSL services on the real servers. In this example, the commands associate real servers rs10 and rs20 with the virtual server.

---

**NOTE:** For clarity, the bindings in the example above are shown as two separate entries. Alternatively, you can enter all the binding information as one command: **bind ssl rs10 ssl rs20 ssl**.

---

#### *USING THE WEB MANAGEMENT INTERFACE*

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of server load balancing option links.
4. Select the Virtual Server link to display the Virtual Server panel.
5. In the Server Name field, enter the name of the virtual server.
6. In the Server IP field, enter the IP address of the virtual server.
7. Click the Add button to add the virtual server to the device's running-config file.
8. Click the Virtual Server Port link at the bottom of the panel to display the Virtual Server Port panel.

9. Click the Modify button for the virtual server you just created. The following panel is displayed:

The screenshot shows a web browser window titled "http://209.157.22.1/ - Microsoft Internet Explorer". The address bar shows "http://209.157.22.1/". The browser displays the Foundry Networks configuration interface. On the left is a navigation tree with the following items: ServerIron, Monitor, Configure, System, Port, VLAN, STP, Trunk, Static Station, SLB (selected), General, Backup, Bind, Real Server, Real Server Port, Router Interface, Source IP, TCP/UDP Port, Virtual Server, Virtual Server Port, TCS, FWLB, GSLB, Web Switching, and Command. The main content area is titled "Virtual Server Port" and contains the following configuration fields:

- Server Name: ssVIP
- TCP/UDP Port: SSL (dropdown menu), User Define (button)
- Status: ☐ Disable ☒ Enable
- Sticky: ☒
- Concurrent: ☐
- Translate: ☒
- DSR: ☐
- HTTP Parameters section:
  - URL Switching: ☐
  - URL Map: None (dropdown menu)
  - URL Hashing: ☐
  - URL Segment Hashing: ☐
  - Cookie Switching: ☐
  - Cookie Name: (text field)
  - Cookie Hashing: ☐
- SSL Parameter section:
  - Session-id Switching: ☒

Below the configuration fields are buttons: Add, Modify, Delete, and Reset. Below these buttons is a [Show] link. At the bottom of the panel are two rows of links: [Host Id][URL Segment][URL Map][Track][Virtual Server][Real Server][Real Server Port][Bind] and [Home][Site Map][Logout][Save][Frame Enable][Disable][TELNET]. The status bar at the bottom shows "http://209.157.22.1/L4VSPpg.htm" and "Internet".

10. Select SSL from the TCP/UDP Port pulldown menu.
11. Click the checkbox next to Session-id Switching.
12. Click the Bind link at the bottom of the panel. The Bind panel is displayed.
13. Select the virtual server from the Virtual Server Name pulldown menu.
14. Select SSL from the Virtual TCP/UDP Port pulldown menu.
15. From the Real Server Name pulldown menu, select a real server that you are binding to this virtual server.
16. Click Bind to bind the real server to the virtual server.
17. Repeat Steps 14 – 16 for each real server you want to bind to this virtual server.
18. Select the Save link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Setting the SSL Aging Period

By default, the ServerIron keeps the entry associating a session\_id with a real server in its database for 30 minutes. After 30 minutes, the entry ages out of the database. If you want to change the length of time the ServerIron keeps the entry in the database, use one of the following methods.

### USING THE CLI

To change the aging period from its default of 30 minutes to 10 minutes:

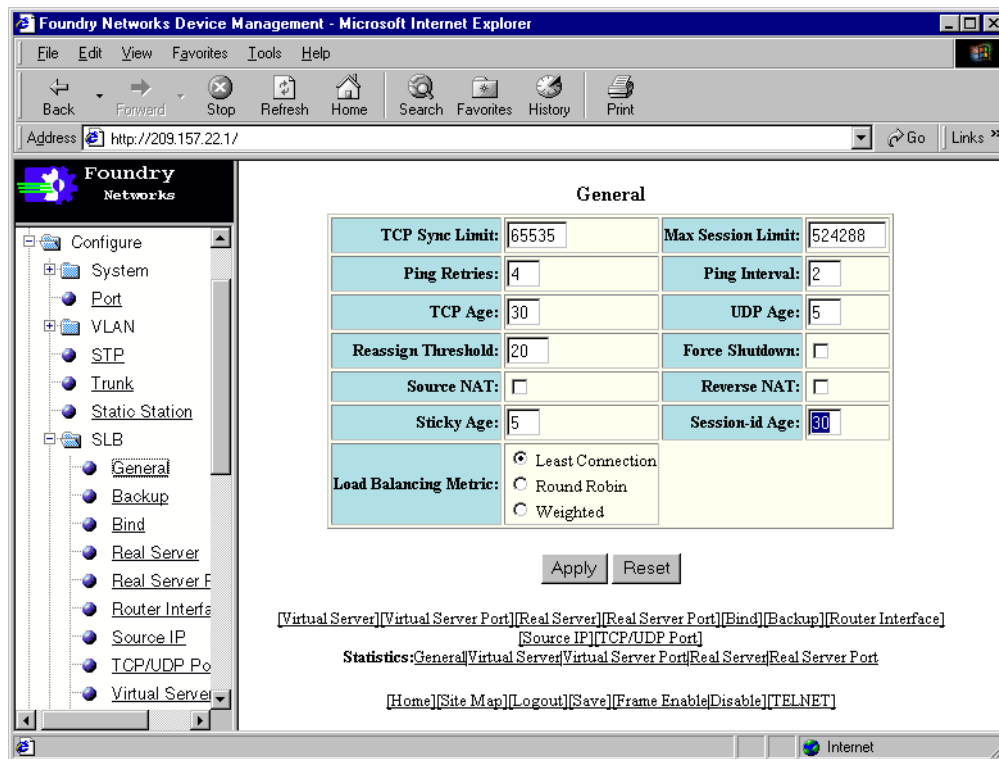
```
ServerIron(config)# server session-id-age 10
```

**Syntax:** server session-id-age <minutes>

The aging period can range from 2 minutes up to 60 minutes.

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of server load balancing option links.
4. Select the General link to display the following panel:



5. In the Session-id Age field, enter a new value for the Session ID aging period. The Session ID aging period can range from 2 minutes up to 60 minutes. The default is 30 minutes.
6. Click the Apply button to save the changes to the device's running-config file.
7. Select the Save link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Setting the Maximum Number of Database Entries

By default, the ServerIron can store in its database 8,192 entries associating a session\_id with a real server. To change the maximum number of database entries, enter the following command:

### USING THE CLI

To change the maximum number of database entries from 8,192 to 64,000:

```
ServerIron(config)# server max-ssl-session-id 64000
```

**Syntax:** server max-ssl-session-id <number>

On the ServerIronXL and ServerIronXL/G, the number of database entries can range from 8,192 to 64,000. On the ServerIron 400 and ServerIron 800, the number of database entries can range from 8,192 to 256,000.

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of server load balancing option links.
4. Select the General link to display the following panel:

**General**

TCP Sync Limit:	65535	Max Session Limit:	524288
Ping Retries:	4	Ping Interval:	2
TCP Age:	30	UDP Age:	5
Reassign Threshold:	20	Force Shutdown:	<input type="checkbox"/>
TCP syn-def:	0	Clock Scale:	0
Backup preference:	5	Backup timer:	10
ICMP message:	<input type="checkbox"/>	L4 check:	<input checked="" type="checkbox"/>
Source NAT:	<input type="checkbox"/>	Reverse NAT:	<input type="checkbox"/>
Sticky Age:	5	Session-id Age:	30
Max ssl session id:	8192	Max URL switch:	100000
Load Balancing Metric:	<input checked="" type="radio"/> Least Connection <input type="radio"/> Round Robin <input type="radio"/> Weighted		

5. In the Max ssl session id field, enter the maximum number of database entries.
6. Click the Apply button to save the changes to the device's running-config file.
7. Select the Save link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Viewing Layer 7 Switching Details and Statistics

You can view the following Layer 7 switching details and statistics:

- URL switching policy information – see “Displaying URL Switching Policy Information” on page 11-47
- Layer 7 switching statistics – see “Displaying Layer 7 Switching Statistics” on page 11-49

### Displaying URL Switching Policy Information

You can display URL switching policy information using either of the following methods.

## USING THE CLI

To display information about a URL switching policy configured on the ServerIron, enter a command such as the following at any level of the CLI:

```
ServerIron#show policy-map p1
Current Policy: 3          Created: 8          Deleted: 5
Table slot 210
-----
Name          : p1                Valid       : Yes
Tree root     : Yes              Method       : prefix

Key           Type                Data
---          -
default      Map Policy          p2
/home        Group ID             1
```

**Syntax:** show policy-map [<policy-map-name>]

This display shows the following information about a URL switching policy.

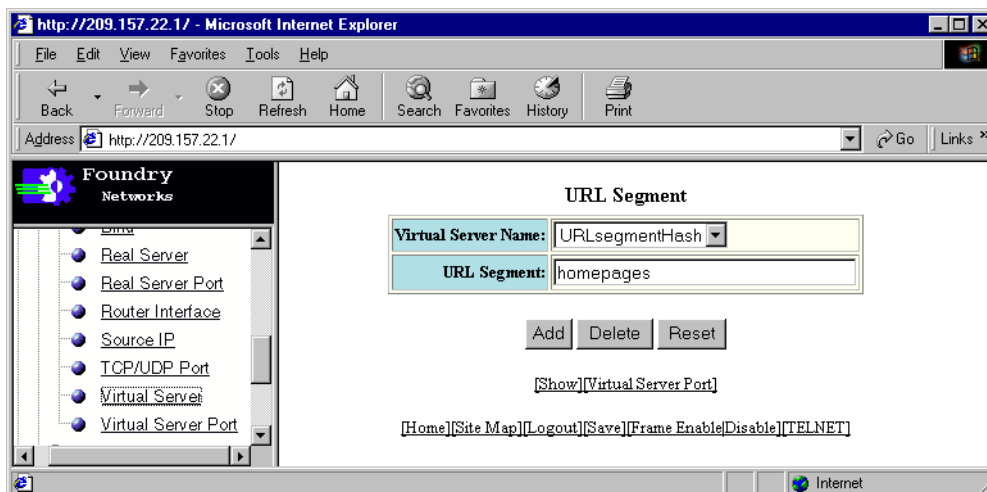
**Table 11.1: URL Switching Policy Information**

This Field...	Displays...
Current Policy	The number of URL switching policies in effect on the ServerIron
Created	The total number of URL switching policies that have been created
Deleted	The number of URL switching policies that have been deleted
Name	The name of the URL switching policy
Valid	Whether the internal structure of the policy is valid
Tree root	Whether the policies that are linked to by this policy have been allocated
Method	The matching method in effect for this policy – either prefix, suffix, or pattern
Key	Either "default" or the selection criteria in effect for this policy
Type	What happens when the selection criteria is met. This can one of the following: Map Policy    The URL string is sent to another URL switching policy for additional matching Group ID      The HTTP request is sent to a server group
Data	Either a server group ID or the name of a URL switching policy

## USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to Web Switching in the tree view to expand the list of Layer 7 switching option links.

4. Select the URL Map link to display the following panel:



This panel displays the following information about the URL switching policies configured on the ServerIron.

**Table 11.2: URL Switching Policy Information**

This Field...	Displays...	
Name	The name of the URL switching policy	
Method	The matching method in effect for this policy – either prefix, suffix, or pattern	
Default	What happens when the URL string does not meet any of the policy's selection criteria: <ul style="list-style-type: none"> <li>If the HTTP request is to be directed to a real server group, the ID of the server group is displayed</li> <li>If the URL string is to be matched against another URL switching policy, the name of the policy is displayed</li> </ul>	
Match	The selection criteria in the URL switching policy. For each set of selection criteria, the following is displayed:	
	URL String Segment	The text of the URL string that the policy looks for
	Group Id/URL Map	What happens when the URL String Segment is found: <ul style="list-style-type: none"> <li>If the HTTP request is to be directed to a real server group, the ID of the server group is displayed</li> <li>If the URL string is to be matched against another URL switching policy, the name of the policy is displayed</li> </ul>

## Displaying Layer 7 Switching Statistics

You can display Layer 7 switching statistics using either of the following methods.

*USING THE CLI*

To display Layer 7 switching statistics, enter the following command at any level of the CLI:

```
ServerIron#show server proxy
  Slot alloc      =          0  Curr free slot      =      99999
  Slot freed      =          0  Slot alloc fail   =          0
  Pkt stored      =          0  Max slot alloc   =          0
  Pkt freed       =          0  Fwd Stored pkt   =          0
  Session T/O     =          0  Sess T/O pkt free =          0
  Session del     =          0  Sess del pkt free =          0
  DB cleanup cnt  =          0  DB cleanup pkt free =          0
  Serv RST to SYN =          0  Send RST to C    =          0
  URL not in 1st pkt =          0  Cookie not in 1st pk =          0
  URL not complete =          0  Cookie not complete =          0
  Sess T/O rev Sess 0 =          0  Sess T/O Sess diff =          0
  Dup SYN Sess diff =          0
  Curr slot used  =          0  Curr pkt stored   =          0
```

**Syntax:** show server proxy

This display shows the following information.

**Table 11.3: Layer 7 Switching Statistics**

This Field...	Displays...
Slot alloc	Number of proxies allocated
Curr free slot	Number of proxies possible
Slot freed	Number of proxies finished
Slot alloc fail	Number of proxy allocation failures
Pkt freed	Number of packets stored by proxy
Max slot alloc	Maximum number of concurrent proxies
Pkt freed	Number of packets freed by proxy
Fwd Stored pkt	Number of stored packets sent to server
Session T/O	Number of session timeouts
Sess T/O pkt free	Number of stored packets freed due to session timeout
Session del	Number of sessions freed by proxy
Sess del pkt free	Number of stored packets deleted when session was freed
DB cleanup cnt	Proxy cleanup count
DB cleanup pkt free	Number of stored packets freed during proxy cleanup
Serv RST to SYN	Number of times the server sent RST to TCP SYN
Send RST to C	Number of times the ServerIron sent RST to client
URL not in 1st pkt	Number of times the URL string was not in the first packet

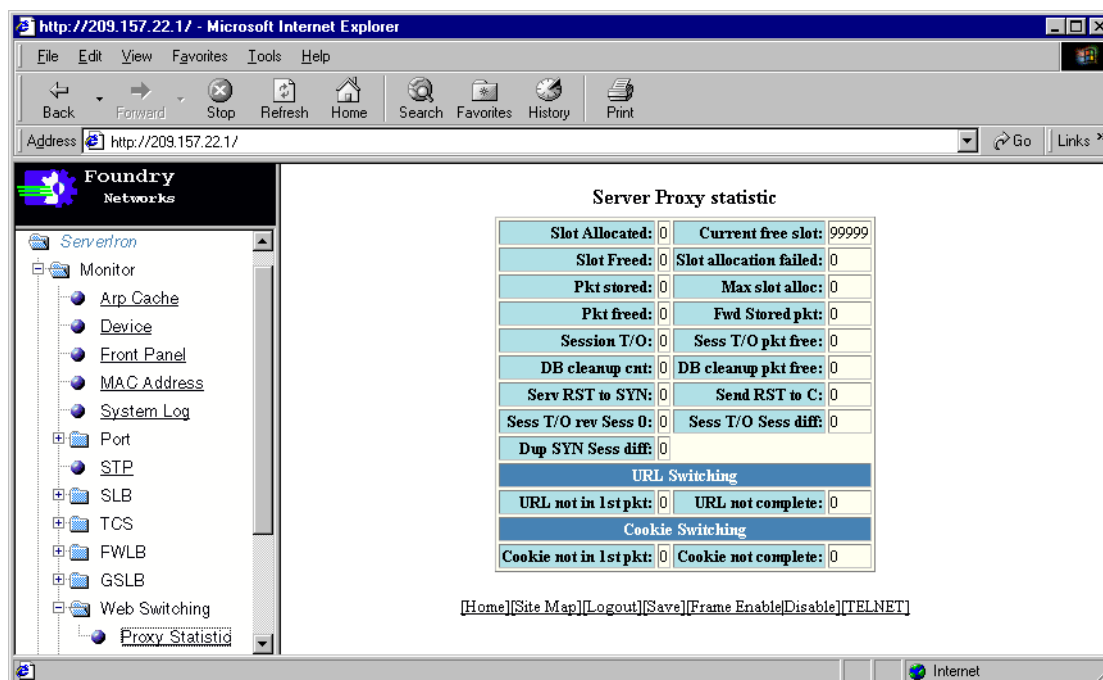


Table 11.3: Layer 7 Switching Statistics (Continued)

This Field...	Displays...
URL not complete	Number of times the URL string was not complete
Cookie not in 1st pk	Number of times the Cookie header was not in the first packet
Cookie not complete	Number of times the Cookie header was not complete
Sess T/O rev Sess 0	Number of session timeouts with no reverse session
Sess T/O Sess diff	Number of session timeouts, internal proxy error
Dup SYN Sess diff	Number of duplicate SYNs received, internal proxy error
Curr slot used	Number of existing proxies
Curr pkt stored	Current number of packets stored by proxy

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read access.
2. Click on the plus sign next to Monitor in the tree view to display the Monitor options.
3. Click on the plus sign next to Web Switching in the tree view to expand the list of Layer 7 switching option links.
4. Select the Proxy Statistic link to display the following panel:



See Table 11.3 on page 11-50 for a description of the items on this panel.

## Setting the Maximum Number of Layer 7 Switching Connections

By default, the ServerIron allows a maximum of 100,000 concurrent Layer 7 switching connections. To change the maximum number of concurrent Layer 7 switching connections, use the server **max-url-switch** command.

### USING THE CLI

To change the maximum number of concurrent Layer 7 switching connections from 100,000 to 160,000:

```
ServerIron(config)# server max-url-switch 160000
```

**Syntax:** server max-url-switch <number>

On the ServerIronXL and ServerIronXL/G, the number of concurrent Layer 7 switching connections can range from 100,000 to 160,000. On the ServerIron 400 and ServerIron 800, the number of concurrent Layer 7 switching connections can range from 100,000 to 512,000.

### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of system configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of server load balancing option links.
4. Select the General link to display the following panel:

**General**

TCP Sync Limit:	65535	Max Session Limit:	524288
Ping Retries:	4	Ping Interval:	2
TCP Age:	30	UDP Age:	5
Reassign Threshold:	20	Force Shutdown:	<input type="checkbox"/>
TCP syn-def:	0	Clock Scale:	0
Backup preference:	5	Backup timer:	10
ICMP message:	<input type="checkbox"/>	L4 check:	<input checked="" type="checkbox"/>
Source NAT:	<input type="checkbox"/>	Reverse NAT:	<input type="checkbox"/>
Sticky Age:	5	Session-id Age:	30
Max ssl session id:	8192	Max URL switch:	100000
Load Balancing Metric:	<input checked="" type="radio"/> Least Connection <input type="radio"/> Round Robin <input type="radio"/> Weighted		

5. In the Max URL switch field, enter the maximum number of concurrent Layer 7 switching connections.
6. Click the Apply button to save the changes to the device's running-config file.
7. Select the Save link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Using HTTP 1.1 for Connections to Real Servers

When the ServerIron receives an HTTP request from a client, it uses its Layer 7 switching configuration to determine to which real server it should send the request. The ServerIron then establishes a TCP connection with the selected real server and sends it the request.

If the request sent from the client to the ServerIron uses HTTP version 1.1, the ServerIron downgrades the HTTP version to 1.0 when it sends the request to the real server. If you want to use HTTP 1.1 for the connection between the ServerIron and the real servers, you can prevent the ServerIron from downgrading the HTTP version to 1.0 by entering the following commands:

```
ServerIron(config)# server virtual-name noDowngrade 192.168.1.234
ServerIron(config-vs-noDowngrade)# port http no-http-downgrade
ServerIron(config-vs-noDowngrade)# exit
```

**Syntax:** port http no-http-downgrade

In HTTP version 1.0 (RFC 1945), a client can send only one request per TCP connection; in HTTP version 1.1 (RFC 2616) a client can send multiple requests per TCP connection. If the ServerIron sends requests to a real server using HTTP 1.1, all the requests in the TCP connection are sent to the same real server that the ServerIron selected using the first request, regardless of the contents of the URL string or Cookie header in the subsequent requests.

## Dropping HTTP Requests when a Server Group Reaches Max Connections

In a Layer 7 switching configuration, policies direct HTTP requests to real servers in load balanced real server groups. When all the real servers in a server group have reached their maximum number of connections (by default, 1,000,000 connections or a threshold set with the **max-conn** command), HTTP requests that would normally go to the server group are instead sent to one of the other real servers bound to the VIP. The ServerIron uses its load balancing metric to select to which of the other real servers it directs the request. If there are no other real servers bound to the VIP besides the ones in the server group, then the request is dropped.

You can change the default behavior so that instead of being sent to a real server bound to the VIP, the requests are dropped. To do this, enter commands such as the following on each real server in the server group:

```
ServerIron(config)# server real-name server1 207.95.7.1
ServerIron(config-rs-server1)# exceed-max-drop
ServerIron(config-rs-server1)# exit
```

**Syntax:** exceed-max-drop

In this example, if server1 reaches its maximum connections threshold, and all the real servers in the server group to which server1 belongs also reach their maximum connections thresholds, HTTP requests that would normally go to server1's server group are dropped.

## Dropping HTTP Requests when a Server Group is Unavailable

By default, if a policy is configured to direct an HTTP request to a server group, but none of the servers in that server group are available, the HTTP request is directed to one of the other server groups configured on the device. You can change this default behavior so that the HTTP request is dropped rather than directed to another server group. To do this, enter the following commands:

```
ServerIron(config)# server virtual-name vip1 192.168.1.234
ServerIron(config-vs-vip1)# port http no-group-failover
ServerIron(config-vs-vip1)# exit
```

**Syntax:** port http no-group-failover



---

# Chapter 12

## Configuring Port and Health Check Parameters

This chapter describes the ServerIron Layer 3, Layer 4, and Layer 7 health checks, how they affect the server and application port states, and how to configure health check parameters. This chapter also describes TCP/UDP application port profiles and how to configure them. See the following sections:

- “Configuring Health Checks”
- “Configuring Session Table Parameters” on page 12-57
- “Configuring the Slow-Start Mechanism” on page 12-62

---

**NOTE:** The health check information in this chapter does not apply to Firewall Load Balancing (FWLB).

---

### Configuring Health Checks

The ServerIron uses Layer 3, Layer 4, and Layer 7 health checks to verify the availability of real servers and applications on the real servers.

When you configure a real server on the ServerIron, the ServerIron sends an ARP request for the real server and then sends an IP ping to the server to verify that the ServerIron can reach the server through the network.<sup>1</sup>

Later, when you bind the real server to a virtual server (VIP), the ServerIron sends a Layer 4 or Layer 7 health check to bring up the port you used for the binding. For example, if you bind a real server to a virtual server using port HTTP, the ServerIron sends an HTTP Layer 7 health check to bring up the HTTP port on the real server.

The ServerIron performs the health checks described above by default. In addition, you can enable periodic Layer 4 or Layer 7 keepalive health checks for individual application ports. Following successful bringup of an application port when you bind a real server to a virtual server, the ServerIron repeats the Layer 4 or Layer 7 keepalive health check to continually verify the health of the port.

The ServerIron selects a Layer 4 or Layer 7 health check based on whether the application port is known to the ServerIron. A Layer 4 health check is a TCP or UDP request and is not related to a specific application. A Layer 7 health check is an application-aware health check designed for the specific application. The following application ports are known to the ServerIron. The ServerIron performs Layer 7 health checks for these ports. For other ports, the ServerIron performs a Layer 4 TCP or UDP health check instead:

---

1. The ARP request is sometimes referred to as a Layer 2 health check since the request is for the real server's hardware layer address.

### TCP Ports

- FTP – the well-known name for port 21. (Ports 20 and 21 both are FTP ports but on the ServerIron, the name “FTP” corresponds to port 21.)
- HTTP – the well-known name for port 80
- IMAP4 – the well-known name for port 143
- LDAP – the well-known name for port 389
- MMS – the well-known name for port 1755
- NNTP – the well-known name for port 119
- PNM – the well-known name for port 7070
- POP3 – the well-known name for port 110
- RTSP – the well-known name for port 554
- SMTP – the well-known name for port 25
- SSL – the well-known name for port 443
- TELNET – the well-known name for port 23

### UDP Ports

- DNS – the well-known name for port 53
- RADIUS – the well-known name for port 1812
- RADIUS-OLD – the ServerIron name for port 1645, which is used in some older RADIUS implementations instead of port 1812

---

**NOTE:** You can add either port 1812 or port 1645 to a given real or virtual server, but you cannot add both ports to the same server.

---

The keepalive health checks are disabled by default. To enable a keepalive health check for an application port, configure a port profile for the port (which automatically enables the keepalive globally for the port) or enable the keepalive on individual real servers that use the port.

## Layer 3 Health Checks

Layer 3 health checks consist of ICMP-based IP pings and ARP requests. When you configure a real server on the ServerIron, the ServerIron sends an ARP request and an IP ping to the real server to verify that the ServerIron can reach the server through the network.

The ServerIron also sends an IP ping to a real server in the following circumstances:

- If the ARP entry for the server times out. In this case, the ServerIron uses the IP ping to create a new ARP entry for the server.<sup>1</sup>
- If the time between the last packet sent to the server and the last packet received from the server increases. In this case, the ServerIron uses the IP ping to determine whether the slowed response time indicates loss of the server. If the server responds to the ping, the ServerIron then sends a Layer 4 or Layer 7 health check, depending on whether the port's application type is known to the ServerIron. The ServerIron sends pings at an interval of 2 seconds apart, and retries unsuccessful pings up to 4 times by default. You can change the ping interval and retries if desired. See “Modifying the Ping Interval and Retries” on page 12-19.

---

1.The ARP request is sometimes referred to as a Layer 2 health check since the request is for the real server's hardware layer address.

## Layer 4 Health Checks

When you bind a real server to a virtual server, the ServerIron performs either a Layer 4 TCP or UDP health check or a Layer 7 health check to bring up the application port that binds the real and virtual servers. If the application port is not one of the applications that is known to the ServerIron, the ServerIron uses a Layer 4 health check. Otherwise, the ServerIron uses the Layer 7 health check for the known application type.

The Layer 4 health check can be a TCP check or a UDP check:

- TCP health check – The ServerIron checks the TCP port's health based on a TCP three-way handshake:
  - The ServerIron sends a TCP SYN packet to the port on the real server.
  - The ServerIron expects the real server to respond with a SYN ACK.
  - If the ServerIron receives the SYN ACK, the ServerIron sends a TCP RESET, satisfied that the TCP port is alive.
- UDP health check – The ServerIron sends a UDP packet with garbage (meaningless) data to the UDP port:
  - If the server responds with an ICMP "Port Unreachable" message, the ServerIron concludes that the port is not alive.
  - If the server does not respond at all, the ServerIron assumes that the port is alive and received the garbage data. Since UDP is a connectionless protocol, the ServerIron and other clients do not expect replies to data sent to a UDP port. Thus, lack of a response indicates a healthy port.

---

**NOTE:** The ServerIron assumes that a port is a UDP port unless you configure the port as a TCP port. To configure a port as a TCP port, add a port profile for the port and specify the port type TCP. See "Configuring a Port Profile" on page 12-21.

---

In addition to performing the health check to bring up the port, the ServerIron repeats the Layer 4 health check if the last packet sent to the server and the last packet received from the server increases. In this case, the ServerIron uses the Layer 4 health check to determine whether the slowed response time indicates loss of the TCP or UDP port.

By default, the ServerIron does not repeat the Layer 4 health check after bringing up the port when you bind the real server to the virtual server. However, you can enable a periodic keepalive health check for the port. To configure the keepalive health check globally, configure a port profile for the port. You also can enable or disable the keepalive health check on individual real servers.

## Layer 7 Health Checks

When you bind a real server to a virtual server using an application port that is known to the ServerIron, the ServerIron sends a Layer 7 health check to the application on the real server to bring up the application port.

Table 12.1 on page 12-4 describes the Layer 7 health checks. Note that for HTTP, DNS, RADIUS, and LDAP health checks, you can configure server-specific health check parameters or use the defaults. See "Configuring Health Check Parameters" on page 12-19. The other Layer 7 health checks are not configurable.

You can enable a Layer 7 health check globally by configuring a port profile or locally by enabling the health check on an individual real server. In addition, you can customize some types of Layer 7 health checks for individual real servers. For example, you can specify a URL that the ServerIron should request on a specific real server when sending the Layer 7 HTTP health check to that server. See "Customizing Layer 7 Health Checks" on page 12-31.

## Health Checking for Real Servers in Other Sub-Nets

The ServerIron must be able to receive the real server's response to a health check in order to assess the success of the health check. In topologies where reply traffic from a real server is guaranteed to pass through the ServerIron, the ServerIron is able to receive replies to the health checks.

However, if the topology is such that the ServerIron and real servers are in different sub-nets or the server reply is not guaranteed to pass back through the ServerIron, you might need to use source NAT and configure a source IP address. Source NAT and source IP addresses allow the ServerIron to have multiple sub-net identities. Generally, the ServerIron is a member of only one sub-net, the sub-net that contains the ServerIron's management IP

address. You can place the ServerIron into up to eight additional sub-nets by enabling source NAT and adding source IP addresses to the ServerIron.

Normally, the ServerIron uses its management IP address as the source address for health check packets. When you enable source NAT and add a source IP address, the ServerIron uses the source IP address as the source for the health check packets. Thus, when the real server replies, the reply is addressed to the source IP address instead of the ServerIron's management IP address.

For an example of how to configure source NAT and source IP addresses, see "Web Hosting with ServerIron and Real Servers in Different Sub-Nets" on page 6-107.

## Health Check Summary

Table 12.1 lists the ServerIron health checks.

**Table 12.1: Health Checks**

Network Layer	Type	When Performed	Description
3	ARP request	<ul style="list-style-type: none"> <li>When you configure a real server</li> </ul>	A standard IP ARP request for the server's MAC address, which the ServerIron adds to its ARP table.
3	IP ping	<ul style="list-style-type: none"> <li>When you configure a real server</li> <li>If the ARP entry ages out</li> <li>If the time between the last packet sent to the server and the last packet received from the server increases</li> </ul>	A standard ICMP-based IP ping.
4	TCP	<ul style="list-style-type: none"> <li>When you bind a TCP application port on a real server to a TCP application port on a virtual server</li> <li>If the time between the last packet sent to the server and the last packet received from the server increases and the server responds to an IP ping</li> <li>At regular intervals, if keepalive is enabled for the port and the port does not have a Layer 7 health check</li> </ul>	<p>The ServerIron attempts to engage in a normal three-way TCP handshake with the port on the real server:</p> <ul style="list-style-type: none"> <li>The ServerIron sends a TCP SYN packet to the port on the real server.</li> <li>The ServerIron expects the real server to respond with a SYN ACK.</li> <li>If the ServerIron receives the SYN ACK, the ServerIron sends a TCP RESET, satisfied that the TCP port is alive.</li> </ul>



Table 12.1: Health Checks (Continued)

Network Layer	Type	When Performed	Description
4	UDP	<ul style="list-style-type: none"><li>• When you bind a UDP application port on a real server to a UDP application port on a virtual server</li><li>• If the time between the last packet sent to the server and the last packet received from the server increases and the server responds to an IP ping</li><li>• At regular intervals, if keepalive is enabled for the port and the port does not have a Layer 7 health check</li></ul>	<p>The ServerIron sends a UDP packet with garbage (meaningless) data to the UDP port.</p> <ul style="list-style-type: none"><li>• If the server responds with an ICMP “Port Unreachable” message, the ServerIron concludes that the port is not alive.</li><li>• If the server does not respond at all, the ServerIron assumes that the port is alive and received the garbage data. Since UDP is a connectionless protocol, the ServerIron and other clients do not expect replies to data sent to a UDP port. Thus, lack of a response is a good outcome.</li></ul>

**Table 12.1: Health Checks (Continued)**

Network Layer	Type	When Performed	Description
7	DNS	<ul style="list-style-type: none"> <li>Immediately following a successful Layer 4 UDP health check</li> <li>At regular intervals, if keepalive is enabled for the port</li> </ul>	<p>The ServerIron performs one or both of the following types of DNS health checks:</p> <ul style="list-style-type: none"> <li>Address-based – The ServerIron sends an address request for a specific domain name. If the server successfully responds with the IP address for the domain name, the server passes the health check.</li> <li>Zone-based – The ServerIron sends a Source-of-Authority (SOA) request for a specific zone name. If the server is authoritative for the zone and successfully responds to the SOA request, the server passes the health check.</li> </ul> <p><b>Note:</b> If you configure both types of DNS health check for a server, the server must successfully respond to both health checks to remain in the server rotation. You enable each type of DNS health check on a global basis and configure them on an individual server basis.</p> <ul style="list-style-type: none"> <li>If the server replies with the requested IP address or zone name, the ServerIron considers the server port to be and marks it ACTIVE.</li> <li>If the server does not reply with the requested IP address or zone name, the ServerIron retries the health check up to the number of times configured (the default is two retries). If the server still does not send the requested information, the ServerIron marks the DNS port on the server FAILED and removes the server from the rotation for DNS services.</li> </ul> <p><b>Note:</b> By default, the health check is non-recursive. If the real server (DNS server) does not successfully reply to the health check, then the DNS port fails the health check. You can enable the real server to perform a recursive lookup for the IP address or zone requested by the health check. In this case, if the real server does not have the requested address or zone, the server can pass the request on to a DNS server with higher authority. See “Recursive Lookups for DNS Health Checks” on page 12-28.</p>

Table 12.1: Health Checks (Continued)

Network Layer	Type	When Performed	Description
7	FTP	<ul style="list-style-type: none"> <li>Immediately following a successful Layer 4 TCP health check</li> <li>At regular intervals, if keepalive is enabled for the port</li> </ul>	<p>The ServerIron waits for a message from the server.</p> <ul style="list-style-type: none"> <li>If the server sends a greeting message with status code 220, the ServerIron resets the connection and marks the port ACTIVE.</li> <li>If the server does not send a greeting message with status code 220, the ServerIron retries the health check up to the number of times configured (the default is two retries). If the server still does not send the expected message, the ServerIron marks the server port FAILED and removes the server from the load-balancing rotation for FTP service.</li> </ul>
7	HTTP (status code)	<ul style="list-style-type: none"> <li>Immediately following a successful Layer 4 TCP health check</li> <li>At regular intervals, if keepalive is enabled for the port</li> </ul>	<p>The ServerIron sends HTTP GET or HEAD requests to cache servers (when using TCS) or HTTP servers (when using SLB).</p> <p>The GET or HEAD request specifies a page (identified by the URL, "Universal Resource Locator") on the server. By default, the ServerIron sends a HEAD request for the default page, "1.0".</p> <ul style="list-style-type: none"> <li>If the server responds with an acceptable status code, the ServerIron resets the connection and marks the port ACTIVE. For SLB, the default acceptable status codes for the check are 200 – 299 and 401. For TCS, the default acceptable status codes are 100 – 499.</li> <li>If the server responds with a different status code, the ServerIron marks the HTTP port FAILED.</li> <li>If the server does not respond, the ServerIron retries the health check up to the number of times configured (the default is two retries). If the server still does not respond, the ServerIron marks the server port FAILED and removes the server from the load-balancing rotation for HTTP service.</li> </ul> <p><b>Note:</b> You can change the status code range for individual servers. If you do so, the defaults are removed and only the status code ranges you specify cause the server to pass the health check.</p>

Table 12.1: Health Checks (Continued)

Network Layer	Type	When Performed	Description
7	HTTP (content verification)	<ul style="list-style-type: none"><li>Immediately following a successful Layer 4 TCP health check</li><li>At regular intervals, if keepalive is enabled for the port</li></ul>	<p>The ServerIron sends HTTP GET or HEAD requests to cache servers (when using TCS) or HTTP servers (when using SLB).</p> <p>The GET or HEAD request specifies a page (identified by the URL) on the server. The ServerIron examines the page and compares the contents of the page to a list of user-defined selection criteria.</p> <p>Based on the results of this comparison, the ServerIron takes one of the following actions with respect to port 80 (HTTP) on the real server.</p> <ul style="list-style-type: none"><li>If the page meets the criteria for keeping the port up, then the ServerIron marks the port ACTIVE. This means that the HTTP application has passed the health check.</li><li>If the page meets the criteria for bringing the port down, then the ServerIron marks the port FAILED.</li><li>If the page meets none of the selection criteria, then the ServerIron marks the port either ACTIVE or FAILED according to a user-defined setting.</li></ul> <p>See “Configuring HTTP Content Matching Lists” on page 12-33 for information on specifying a page to check and setting up lists of selection criteria</p>

Table 12.1: Health Checks (Continued)

Network Layer	Type	When Performed	Description
7	Scripted (content verification health check for unknown ports)	<ul style="list-style-type: none"> <li>Immediately following a successful Layer 4 TCP health check</li> <li>At regular intervals, if keepalive is enabled for the port</li> </ul>	<p>The ServerIron sends a SYN packet to a port on a real server, then waits for the real server to send back a packet in response.</p> <p>The ServerIron looks in the response packet for a user-specified ASCII string, defined in a matching list. The ServerIron compares the contents of the string to a list of user-defined selection criteria in the matching list.</p> <p>Based on the results of this comparison, the ServerIron takes one of the following actions with respect to the port on the real server.</p> <ul style="list-style-type: none"> <li>If the text in the response meets the criteria for keeping the port up, then the ServerIron marks the port ACTIVE.</li> <li>If the text in the response meets the criteria for bringing the port down, then the ServerIron marks the port FAILED.</li> <li>If the text in the response meets none of the selection criteria, then the ServerIron marks the port either ACTIVE or FAILED according to a user-defined setting.</li> <li>If no response is received within the configured interval (the default is five seconds), the ServerIron sends a RST and retries the health check. After the configured number of retries (the default is two retries), if the server still does not respond, the ServerIron marks the server port FAILED.</li> </ul> <p>See "Configuring Scripted Health Checks" on page 12-36 for more information.</p>
7	IMAP4	<ul style="list-style-type: none"> <li>Immediately following a successful Layer 4 TCP health check</li> <li>At regular intervals, if keepalive is enabled for the port</li> </ul>	<p>The ServerIron waits for a message from the IMAP4 server.</p> <ul style="list-style-type: none"> <li>If the server sends a greeting message that starts with "* OK", The ServerIron sends a Logout command to the IMAP4 port on the real server, then resets the connection and marks the port ACTIVE.</li> <li>If the server does not send a greeting message that starts with "* OK", the ServerIron retries the health check up to the number of times configured (the default is two retries). If the server still does not send the expected message, the ServerIron marks the server port FAILED and removes the server from the load-balancing rotation for IMAP4 service.</li> </ul>

**Table 12.1: Health Checks (Continued)**

Network Layer	Type	When Performed	Description
7	LDAP	<ul style="list-style-type: none"> <li>Immediately following a successful Layer 4 TCP health check</li> <li>At regular intervals, if keepalive is enabled for the port</li> </ul>	<p>The ServerIron sends a bind request to the LDAP server and waits for a reply. The bind request includes a configurable version number. The version number can be 2 or 3. The default is 3.</p> <ul style="list-style-type: none"> <li>If the server sends a bind reply with result code 0 (no error), the ServerIron resets the connection and marks the port ACTIVE.</li> <li>If the server does not send a bind reply by the time the LDAP keepalive retries expires, the ServerIron retries the health check up to the number of times configured (the default is two retries). If the server still does not respond, the ServerIron marks the server port FAILED and removes the server from the load-balancing rotation for LDAP service.</li> </ul>
7	MMS	<ul style="list-style-type: none"> <li>Immediately following a successful Layer 4 TCP health check</li> <li>At regular intervals, if keepalive is enabled for the port</li> </ul>	<p>The ServerIron sends an intentionally invalid request to the server.</p> <ul style="list-style-type: none"> <li>If the server replies with a packet containing the value "MMS", the ServerIron marks the port ACTIVE.</li> <li>If the server does not reply with a packet containing the value "MMS", the ServerIron retries the health check up to the number of times configured (the default is two retries). If the server still does not respond, the ServerIron marks the server port FAILED and removes the server from the load-balancing rotation for MMS service.</li> </ul> <p><b>Note:</b> You can view the ServerIron's invalid request in the MMS server log. The log entry has error code 400.</p>
7	NNTP	<ul style="list-style-type: none"> <li>Immediately following a successful Layer 4 TCP health check</li> <li>At regular intervals, if keepalive is enabled for the port</li> </ul>	<p>The ServerIron waits for a message from the NNTP server.</p> <ul style="list-style-type: none"> <li>If the server sends a greeting message with status code 200 or 201, the ServerIron sends a Quit command to the NNTP port on the real server, then resets the connection by sending a quit and a RESET, one immediately after the other, and marks the port ACTIVE.</li> <li>If the server does not send a greeting message with status code 200 or 201, the ServerIron retries the health check up to the number of times configured (the default is two retries). If the server still does not send the expected message, the ServerIron marks the server port FAILED and removes the server from the load-balancing rotation for NNTP service.</li> </ul>

Table 12.1: Health Checks (Continued)

Network Layer	Type	When Performed	Description
7	PNM	<ul style="list-style-type: none"> <li>Immediately following a successful Layer 4 TCP health check</li> <li>At regular intervals, if keepalive is enabled for the port</li> </ul>	<p>The ServerIron sends a PNM file request that does not have a file name.</p> <ul style="list-style-type: none"> <li>If the server sends a reply containing the value "PNA", the ServerIron marks the port ACTIVE.</li> <li>If the server does not send a reply containing the value "PNA", the ServerIron retries the health check up to the number of times configured (the default is two retries). If the server still does not send the expected message, the ServerIron marks the server port FAILED and removes the server from the load-balancing rotation for PNM service.</li> </ul>
7	POP3	<ul style="list-style-type: none"> <li>Immediately following a successful Layer 4 TCP health check</li> <li>At regular intervals, if keepalive is enabled for the port</li> </ul>	<p>The ServerIron waits for a message from the POP3 server.</p> <ul style="list-style-type: none"> <li>If the server sends a greeting message that starts with "+ OK", the ServerIron sends a Quit command to the POP3 port on the real server, then resets the connection by sending a quit and a RESET, one immediately after the other, and marks the port ACTIVE</li> <li>If the server does not send a greeting message that starts with "+ OK", the ServerIron retries the health check up to the number of times configured (the default is two retries). If the server still does not send the expected message, the ServerIron marks the server port FAILED and removes the server from the load-balancing rotation for POP3 service.</li> </ul>

**Table 12.1: Health Checks (Continued)**

Network Layer	Type	When Performed	Description
7	RADIUS	<ul style="list-style-type: none"> <li>Immediately following a successful Layer 4 UDP health check</li> <li>At regular intervals, if keepalive is enabled for the port</li> </ul>	<p>The ServerIron sends an authentication request with a user name, password, and key to the RADIUS server. The account information does not need to be valid for the server to pass the health check. In fact, to prevent someone from learning account information by observing the ServerIron's RADIUS health check, Foundry Networks recommends you use invalid information.</p> <ul style="list-style-type: none"> <li>If the server replies with the result code "ACCEPT" or "REJECT", the ServerIron considers the port to be ok and marks it ACTIVE.</li> <li>If the server does not reply or the server Sends an ICMP "Destination Unreachable" message, the ServerIron retries the health check up to the number of times configured (the default is two retries). If the server still does not reply with "ACCEPT" or "REJECT", the ServerIron marks the RADIUS port FAILED and removes the server from the rotation for RADIUS services.</li> </ul> <p><b>Note:</b> You can configure a check either for the well-known RADIUS port number 1812 or 1645. You cannot configure a health check for both of these ports on the same server.</p>
7	RTSP	<ul style="list-style-type: none"> <li>Immediately following a successful Layer 4 TCP health check</li> <li>At regular intervals, if keepalive is enabled for the port</li> </ul>	<p>The ServerIron sends a standard RSTP option packet, using sequence number 1.</p> <ul style="list-style-type: none"> <li>If the server responds with an acceptable status code, the ServerIron resets the connection and marks the port ACTIVE. For SLB, the default acceptable status codes for the check are 200 – 299 and 401.</li> <li>If the server responds with a different status code, the ServerIron marks the port FAILED.</li> <li>If the server does not respond, the ServerIron retries the health check up to the number of times configured (the default is two retries). If the server still does not respond, the ServerIron marks the server port FAILED and removes the server from the load-balancing rotation for RSTP service.</li> </ul>



Table 12.1: Health Checks (Continued)

Network Layer	Type	When Performed	Description
7	SMTP	<ul style="list-style-type: none"><li>Immediately following a successful Layer 4 TCP health check</li><li>At regular intervals, if keepalive is enabled for the port</li></ul>	<p>The ServerIron waits for a message from the SMTP server.</p> <ul style="list-style-type: none"><li>If the server sends a greeting message with status code 220, the ServerIron sends a Quit command to the SMTP port on the real server, then resets the connection by sending a quit and a RESET, one immediately after the other, and marks the port ACTIVE.</li><li>If the server does not send a greeting message with status code 220, the ServerIron retries the health check up to the number of times configured (the default is two retries). If the server still does not send the expected message, the ServerIron marks the server port FAILED and removes the server from the load-balancing rotation for SMTP service.</li></ul>

**Table 12.1: Health Checks (Continued)**

Network Layer	Type	When Performed	Description
7	SSL ServerIronXL only	<ul style="list-style-type: none"> <li>Immediately following a successful Layer 4 TCP health check</li> <li>At regular intervals, if keepalive is enabled for the port</li> </ul>	<p>The ServerIron initiates an SSL connection with the server on TCP port 443, a secure link is negotiated, and encrypted data is transferred across it.</p> <p>After the SSL connection is established, the ServerIron sends the SSL server an HTTP GET or HEAD request. The GET or HEAD request specifies a page containing the URL of a page on the server. By default, the ServerIron sends a HEAD request for the default page, "1.0", although this can be changed with the <b>port ssl url</b> command.</p> <ul style="list-style-type: none"> <li>If the server responds with an acceptable status code, the ServerIron resets the connection and marks the port ACTIVE.</li> <li>If the server does not respond, the ServerIron retries the health check up to the number of times configured (the default is two retries). If the server still does not respond, the ServerIron marks the server port FAILED and removes the server from the load-balancing rotation for SSL service.</li> </ul> <p><b>Note:</b> You can configure the ServerIronXL to use the SSL health check method used by other ServerIron models. See below and "Using Simple SSL Health Checks (ServerIronXL Only)" on page 12-30.</p>
7	SSL other ServerIron models	<ul style="list-style-type: none"> <li>Immediately following a successful Layer 4 TCP health check</li> <li>At regular intervals, if keepalive is enabled for the port</li> </ul>	<p>The ServerIron sends an SSL client hello with the SSL SID set to 0:</p> <ul style="list-style-type: none"> <li>If the server responds, then the ServerIron resets the connection and marks the port ACTIVE.</li> <li>If the server does not respond, the ServerIron retries the health check up to the number of times configured (the default is two retries). If the server still does not respond, the ServerIron marks the server port FAILED and removes the server from the load-balancing rotation for SSL service.</li> </ul>

Table 12.1: Health Checks (Continued)

Network Layer	Type	When Performed	Description
7	Telnet	<ul style="list-style-type: none"> <li>Immediately following a successful Layer 4 TCP health check</li> <li>At regular intervals, if keepalive is enabled for the port</li> </ul>	<p>The ServerIron waits for a message from the Telnet server.</p> <ul style="list-style-type: none"> <li>If the server sends a command string that starts with the IAC escape characters ("FF"), the ServerIron resets the connection and marks the port ACTIVE.</li> <li>If the server does not send a command that starts with the IAC escape character, the ServerIron retries the health check up to the number of times configured (the default is two retries). If the server still does not send the expected escape character, the ServerIron marks the server port FAILED and removes the server from the load-balancing rotation for Telnet service.</li> </ul>

## Server and Application Port States

The ServerIron displays state information for real servers and application ports. The state of a server or application port is based on the results of health checks on that server or port.

### Server States

Table 12.2 lists the server states.

Table 12.2: Server States

State	Description
ENABLED	There is no link to the real server. The real server is configured on the ServerIron but is not physically connected to the ServerIron.
FAILED	The real server has failed to respond to repeated Layer 3 health checks (IP pings). Typically, a real server changes to the FAILED state from the SUSPECT state.

**Table 12.2: Server States (Continued)**

State	Description
TEST	<p>The real server is still reachable at Layer 3, but at least one of the application ports on the server has failed to respond to its health checks. If the application port is not a TCP/UDP port known to the ServerIron or the Layer 7 health check for the port is disabled, only the Layer 4 health check is used. If the service is a TCP or UDP port known to the ServerIron and the Layer 7 health check is enabled, then the application must pass both health checks to avoid entering the TEST state.</p> <p>The ServerIron continues to try to reach the application indefinitely. Thus, if the server continues to be reachable at Layer 3, the state will remain TEST so long as the ServerIron cannot reach the application that is failing its health check.</p> <p><b>Note:</b> The ServerIron rotates the health checks among the load-balanced servers. As a result, you may see a server's state repeatedly change from TEST to FAILED and then back to TEST. When the ServerIron is testing the service on that server, the state is TEST. When the ServerIron rotates to another server, the state is FAILED until the ServerIron rotates around to the server again, at which time the state changes back to TEST.</p>
SUSPECT	<p>The ServerIron associates a time stamp with each packet sent to and received from the real servers. If the time gap between the last packet received from the server and the last packet sent to the server grows to 3 or 4 seconds, the ServerIron sends a ping (Layer 3 health check) to the server. If the server doesn't respond within the ping interval (a configurable parameter), the ServerIron changes the state to SUSPECT and resends the ping, up to the number of retries specified by the ping retries parameter (also configurable). If the server still doesn't respond after all the retries, the state changes to FAILED. If the server does respond, the state changes to ACTIVE.</p>
GRACE_DN	<p>The forced-shutdown option has been used to gracefully shut the server down.</p>
ACTIVE	<p>The server has responded to the Layer 3 health check (IP ping). Also, all the services on the real server have passed their Layer 4 (and if applicable, Layer 7) health checks.</p>

## Application Port States

Table 12.3 lists the application port states.

**Table 12.3: Application Port States**

State	Description
ENABLED	<p>There is no link to the server. The server is configured on the ServerIron but is not connected to the ServerIron. (This is the same as the ENABLED server state.)</p>
FAILED	<p>The application has failed to respond to repeated Layer 4 or (if applicable) Layer 7 health checks. Typically, an application changes to the FAILED state from the SUSPECT state. Note that if a application does not pass the Layer 4 health check, the ServerIron does not waste resources on the Layer 7 health check, since the application clearly is not available. When an application enters the FAILED state, the state of the real server itself moves to the TEST state while the ServerIron continually tries to reach the failed application.</p>

Table 12.3: Application Port States (Continued)

State	Description
TEST	The server is still reachable at Layer 3, but the application has failed to respond to its Layer 4 (or if applicable, Layer 7) health check.
SUSPECT	The ServerIron associates a time stamp with each packet sent to and received from the real servers. If the time gap between the last packet received from the server and the last packet sent to the server grows to 3 or 4 seconds, the ServerIron sends a Layer 4 health check to the service. (If applicable, and if the server passes the Layer 4 health check, the ServerIron then sends a Layer 7 health check to the application.) If the application doesn't respond within a specified interval, the ServerIron changes the state to SUSPECT and resends the Layer 4 (and if applicable, Layer 7) health check up to a specific number of retries. If the application still doesn't respond after all the retries, the state changes to FAILED and the server state changes to TEST. If the application does respond, the application state changes to ACTIVE.
GRACE_DN	The forced-shutdown option has been used to gracefully shut the server down.
ACTIVE	The application has passed its Layer 4 (and if applicable, Layer 7) health check.
UNBND	The application is configured on the real server but is not yet bound to a virtual server.

The following sections describe how to display the state information.

### Displaying Real Server State Information

To display state information for real servers and their application ports, use either of the following methods. For complete information about these displays, see "Displaying Real Server Information" on page 6-74.

#### USING THE CLI

To display real server information, enter the following command at any level of the CLI:

```
ServerIron(config)# show server real
Real Servers Info

Server State - 1:enabled, 2:failed, 3:test, 4:suspect, 5:grace_dn, 6:active
Name:rs1          IP:      207.95.7.1:1    State:1    Wt:1      Max-conn:1000000
Src-nat (cfg:op) = 0: 0 Dest-nat-(cfg:op) = 0: 0
Remote server: No      Dynamic: No  :Mac-info: ffff
Port  State   Ms  CurConn  TotConns  Rx-pkts  Tx-pkts  Rx-octet  Tx-octet  Reas
http  enabled  0    0         0         0         0         0         0         0
      Keepalive(G/L:Off/Off):Off
      Status Code(s):  default (200-299, 401)
      HTTP URL: "HEAD /"
defaultunbnd  0    0         0         0         0         0         0         0
Server  Total      0    0         0         0         0         0         0
```

information for remaining real servers omitted for brevity...

The state information shown by this display is highlighted in bold type in the example above. The state of the server itself is listed first, then the states of each of the application ports configured on the server are displayed.

In this example, the server itself is enabled. The HTTP port also is enabled, but the "default" port (a port the ServerIron automatically configures on all the real and virtual servers) is unbound. These states are typical of a ServerIron that is configured for deployment but has not been connected to the real servers.

The information under the row for the HTTP application shows settings for the Layer 7 health checks for the port. For information about HTTP health checks and other configurable Layer 7 health check parameters, see “Customizing Layer 7 Health Checks” on page 12-31.

#### [USING THE WEB MANAGEMENT INTERFACE](#)

To display information for a real server configured on the ServerIron, do one of the following:

- Select the [Real Server](#) link from the list of statistics links at the bottom of the General panel or other SLB panels that have the links.
- Click on the plus sign next to Monitor in the tree view, then the plus sign next to SLB, then the [Real Server](#) link.

---

**NOTE:** If data entry fields for a real server are displayed instead of a table listing real servers, then there are no real servers configured on the ServerIron.

---

### Displaying Virtual Server State Information

To display state information for virtual servers and their application ports, use either of the following methods. For complete information about these displays, see “Displaying Virtual Server Information” on page 6-82.

#### [USING THE CLI](#)

To display virtual server information, enter the following command at any level of the CLI:

```
Virtual Servers Info
Server Name: v100          IP : 209.157.23.100 : 4
Status: enabled Predictor: least-conn TotConn: 4233
Dynamic: No HTTP redirect: disabled
Sym: group = 1 state = 5 priority = 2 keep = 0
Activates = 4, Inactive= 3
Port  State      Sticky  Concur      CurConn  TotConn  PeakConn
http  enabled    NO      NO           0        4233     39
default enabled  NO      NO           0         0        0
```

*information for remaining virtual servers omitted for brevity...*

In this example, the virtual server and the application ports configured on the server are enabled, indicating that the server and the application ports are configured on the ServerIron but the ServerIron is not connected to the real servers bound to this virtual server. See “Displaying Real Server State Information” on page 12-17 for descriptions of the server and application states.

---

**NOTE:** The number following “state” in the “Sym” row indicates the Symmetric SLB state of this VIP. See “Displaying Virtual Server Information” on page 6-82.

---

#### [USING THE WEB MANAGEMENT INTERFACE](#)

To display information for a virtual server configured on the ServerIron, do one of the following:

- Select the [Virtual Server](#) link from the list of statistics links at the bottom of the General panel or other SLB panels that have the links.
- Click on the plus sign next to Monitor in the tree view, then the plus sign next to SLB, then the [Virtual Server](#) link.

---

**NOTE:** If data entry fields for a virtual server are displayed instead of a table listing virtual servers, then there are no virtual servers configured on the ServerIron.

---

## Reassign Threshold

The **reassign threshold** specifies the number of contiguous inbound TCP-SYN packets a real server can fail to respond to before the ServerIron changes the application state to FAILED and the server state to TEST. The default reassign threshold is 21. The server and application states are described in “Server and Application Port States” on page 12-15.

If the field reaches the reassign threshold, the ServerIron marks the application failed. The value of an application’s Reas field is reset to 0 when the ServerIron receives a TCP SYN ACK from the application. No other type of traffic can clear this field.

If a real server seems to be triggering the reassign threshold too frequently, you can increase the reassign threshold. The default is 21 and the range of values is 6 – 254. This is a global parameter. See “Modifying the Reassign Threshold” on page 12-29.

### Notes Regarding the Reassign Threshold

- It is possible to take a service down without triggering the reassign threshold. For example, in a lab environment where the server is not receiving TCP SYNs, the service might be down but since the ServerIron is not sending new requests to the server, the server does not fail to respond to enough consecutive TCP SYNs to meet the reassign threshold. As a result, the ServerIron indicates the server and the service are ACTIVE when in fact they are offline.
- The ServerIron does not try to reassign the client’s request to another server if you configure the application port to be sticky. The sticky option configures the ServerIron to override load-balancing and send all client requests for the application to the same server during a given session.
- The reassign threshold counter is not incremented in SwitchBack (Direct Server Return) configurations.

## Configuring Health Check Parameters

This section describes how to configure the following health check parameters:

- IP ping interval and retries
- Layer 3 health check when adding real servers
- Layer 4 health check state (enabled or disabled)
- Port profile parameters
- Reassign threshold
- Health-checking procedure used in releases prior to 07.1.05
- SSL health check method

### Modifying the Ping Interval and Retries

The ServerIron automatically uses a Layer 3 health check consisting of ICMP echo requests (pings) to check the health of a real server. Ping is enabled by default and cannot be disabled. However, you can modify the ping interval and number of retries.

The ping interval can be from 1 – 10 seconds. The default is 2 seconds. The number of ping retries can be from 2 – 10. The default retry value is 4.

#### USING THE CLI

To modify the interval between ping retries to 8 seconds from the default value of 2 seconds, enter the following command:

```
ServerIron(config)# server ping-interval 8
```

**Syntax:** server ping-interval <1-10>

You can specify from 1 – 10 seconds.

To modify the number of times the ServerIron will ping a real server before changing the server state to FAILED, enter a command such as the following:

```
ServerIron(config)# server ping-retries 7
```

**Syntax:** server ping-retries <2-10>

#### ***USING THE WEB MANAGEMENT INTERFACE***

To modify the number of times a server is pinged by the ServerIron before the ServerIron changes the server state to FAILED:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu. The panel shown in Figure 6.8 on page 6-22 will appear.
5. Enter a value from 2 – 10 in the Ping Retries field.
6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

To modify the interval length between ping retries:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu. The panel shown in Figure 6.8 on page 6-22 will appear.
5. Enter a value from 1 – 10 in the Ping Interval field.
6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

#### **Disabling the Layer 3 Health Check for Real Servers**

By default, when you add a real server configuration to the ServerIron, the ServerIron uses a Layer 3 health check (IP ping) to determine the server's reachability. If the real server responds to the ping, the ServerIron changes the server's state to ACTIVE and begins using the server for client requests.

You can globally disable the Layer 3 health check for local servers or remote servers. You also can disable the Layer 3 health check on individual real servers. When you disable the Layer 3 health check, the ServerIron sends an ARP request for the default gateway and makes the server's state ACTIVE as long as the ARP entry is present in the ServerIron's ARP cache.

##### ***Disabling the Layer 3 Health Check for Local Real Servers***

To globally disable the Layer 3 health check for all local real servers, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# server no-real-l3-check
```

**Syntax:** [no] server no-real-l3-check

##### ***Disabling the Layer 3 Health Check for Remote Real Servers***

To globally disable the Layer 3 health check for all remote real servers, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# server no-remote-l3-check
```

**Syntax:** [no] server no-remote-l3-check



### **Disabling the Layer 3 Health Check for an Individual Real Server**

To disable the Layer 3 health check on an individual real server, enter the following command at the configuration level for the server:

```
ServerIron(config-rs-R1)# no-l3-check
```

**Syntax:** [no] no-l3-check

This command applies to local real servers and remote real servers.

### **Disabling or Re-Enabling Layer 4 Health Checks**

You can globally disable or re-enable Layer 4 TCP or UDP health checks for servers. The Layer 4 health checks are enabled by default. Disable the Layer 4 health checks if you are configuring the ServerIron to load balance traffic to multiple servers on the other side of routers and you want to load-balance the traffic according to TCP or UDP application. If you do not disable the health checks in this type of configuration, the routers will fail the health checks (because the target applications for the health checks are not on the routers themselves) and the ServerIron will stop forwarding traffic to those servers.

---

**NOTE:** If you are using the ServerIron to load-balance TCP and UDP traffic through routers, you also must add each router as a real server and disable the HTTP port on each of the real servers. HTTP is enabled by default on all real servers.

---

#### **USING THE CLI**

To disable Layer 4 health checks, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# no server l4-check
```

**Syntax:** [no] server l4-check

To re-enable Layer 4 health checks, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# server l4-check
```

#### **USING THE WEB MANAGEMENT INTERFACE**

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu.
5. Select the checkbox next to L4 check to remove or replace the checkmark in the box. When the box contains a checkmark, Layer 4 health checks are enabled. If the box does not contain a checkmark, Layer 4 health checks are disabled.
6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### **Configuring a Port Profile**

A **port profile** is a set of attributes that globally define an application port. Once defined, the port has the same attributes on all the real and virtual servers that use the port. Port profiles are useful if you want to globally change the attributes of a port known to the ServerIron (see the list in "Customizing Layer 7 Health Checks" on page 12-31) or you want to globally define a port that is not known to the ServerIron. You also can specify or change port attributes locally, on the Real Server and Virtual Server configuration levels.

If you want to enable the keepalive health check for an application port, you must configure a port profile for the port.

Table 12.4 lists the port attributes you can configure at the port profile level.

**Table 12.4: Port Profile Attributes**

Attribute	Description
Port type (TCP or UDP)	<p>This attribute applies only to ports for which the ServerIron does not already know the type. For example, if a real server uses port 8080 for HTTP (a TCP port), you can globally identify 8080 as a TCP port. The ServerIron assumes that ports for which it does not know the type are UDP ports.</p> <p><b>Note:</b> To display a list of the ports for the ServerIron already knows the type, enter the <b>server port ?</b> command at the global CONFIG level of the CLI.</p>
Keepalive interval and retries	<p>The number of seconds between health checks and the number of times the ServerIron re-attempts a health check to which the server does not respond. You can specify from 2 – 120 seconds for the interval. You can specify from 1 – 5 retries.</p>
Keepalive state	<p>Whether the ServerIron's health check for the port is enabled or disabled. Recurring Layer 4 and Layer 7 health checks are disabled by default. When you configure a port profile, the software automatically globally enables the health check for the application. You also can explicitly disable or re-enable the keepalive health check at this level.</p> <p><b>Note:</b> If you are configuring a port profile for a port that is known to the ServerIron, the keepalive parameters affect Layer 7 health checks. For other ports, the keepalive parameters affect Layer 4 health checks.</p>
Keepalive port	<p>By default, the ServerIron bases the health of an application port on the port itself. You can specify a different application port for the health check. In this case, the ServerIron bases the health of an application port on the health of the other port you specify.</p> <p><b>Note:</b> You cannot base the health of a port well-known to the ServerIron on the health of another port, whether the port is well-known or not well-known.</p>
Source of health for alias port	<p>By default, the ServerIron performs independent health checks on an alias port and its master port. You can configure the ServerIron to base the health of an alias port on the state of its master port.</p>
TCP or UDP age	<p>The number of minutes a TCP or UDP session table entry can remain inactive before the ServerIron times out the entry. This parameter is set globally for all TCP or UDP ports but you can override the global setting for an individual port by changing that port's profile. You can set the TCP or UDP age from 2 – 60 minutes. The default TCP age is 30 minutes. The default UDP age is five minutes.</p> <p><b>Note:</b> Since UDP is a connectionless protocol, the ServerIron does not remove a UDP session from its session table until the session times out. TCP is a connection-based protocol. Thus, for TCP sessions, the ServerIron removes the session as soon as the client or server closes the session.</p> <p><b>Note:</b> The ServerIron immediately deletes a UDP DNS or RADIUS session table entry when the ServerIron receives a reply for the application from a real server. If desired, you can configure the ServerIron to age these ports like other UDP ports, using the UDP age timer. See "Normal UDP Aging for DNS and RADIUS" on page 6-60.</p>

**Table 12.4: Port Profile Attributes (Continued)**

Attribute	Description
Session synchronization	In Symmetric SLB configurations, this attribute provides failover for individual sessions on the application port. Normally, existing sessions are not carried over from one ServerIron to another during failover. See "Using Symmetric Server Load Balancing" on page 7-1.
Connection logging	You can enable logging for session table entries created for this port. See "Enabling Syslog Messages for Session Table Entries" on page 12-60.
Slow start	Configures the ServerIron to control the rate of new connections to the application port to allow the server to ramp up. See "Port Slow-Start Mechanism" on page 12-64.
Smooth factor	If you plan to use server response time as a load-balancing method, you can adjust the amount of preference the ServerIron gives the most recent response time compared to the previous response time.
Recursive DNS health checks	By default, a Layer 7 health check for a DNS port sends the query only to the real server (DNS server). If the DNS server does not reply with the IP address or zone name requested by the health check, the port fails the health check.  You can enable the real server to perform a recursive lookup for the IP address or zone requested by the health check of the well-known DNS port (53).

You also can change port attributes locally, on the Real Server and Virtual Server configuration levels. Port profiles simplify configuration by enabling you to characterize a port globally. For example, if many of your real servers use TCP port 80 (the well-known number for HTTP) and you want to change the keepalive interval for the port, you can do so globally. You do not need to change the value multiple times on each real server.

The ServerIron knows the port types of some well-known port numbers. If you are using a port number for which the ServerIron does not know the port type, you can specify whether the port is TCP or UDP and configure its keepalive values globally. You do not need to define the port on every server.

**NOTE:** Unless a port is known to the ServerIron to be a TCP port, the ServerIron assumes the port is UDP. If you are using a port number that is not known to the ServerIron and the port type is TCP, you must specify this either globally (using a port profile) or locally (when configuring the individual real servers and virtual servers). Otherwise, the ServerIron will use a UDP health check to test the port and the port will fail the health check.

**NOTE:** If you bind an application port on a real server to the same port on a virtual server, the port on the real server inherits the attributes of the port on the virtual server.

#### ***Adding a TCP or UDP Port, Specifying the Port Type, and Configuring the Keepalive Health Check***

For an application port that is not known to the ServerIron, the ServerIron assumes that the port is a UDP port. In addition, the ServerIron does not perform keepalive health checks for the port. You can configure a port profile for the port and specify whether the port is TCP or UDP and also set keepalive health check parameters for the port.

Even for ports that are known to the ServerIron, you must configure a profile for the port to globally configure the port's parameters and configure the keepalive health check. After you add the port by indicating whether it is a TCP or UDP port, the ServerIron automatically enables the keepalive health check for the port.

**NOTE:** Enabling or disabling a keepalive health check does not affect the health check the ServerIron sends when you bind a real server to a virtual server using the application port. The keepalive health check state also does not affect the health checks the ServerIron sends if the server's response time slows.

The keepalive interval and retry values for each type of TCP/UDP health check are global parameters. For example, if you change the number of retries for the HTTP health check (TCP port 80), the change applies to all instances of port 80 on all the real servers configured on the ServerIron.

---

**Table 12.5: Keepalive Health Check States**

State		Effect
Global (entire ServerIron)	Local (specific real server)	
Disabled	Disabled	Health check is disabled
Disabled	Enabled	Health check is enabled
Enabled	Disabled	Health check is enabled
Enabled	Enabled	Health check is enabled

As shown in this table, once a keepalive health check is enabled, to disable it you must do so both globally and locally. If you want to enable keepalive health checks only on specific real servers (locally), you can easily do so by making sure the health checks are disabled globally, then enabling them on individual real servers.

To enable or disable a keepalive health check globally, use one of the following methods. To enable or disable a keepalive health check locally, see “Locally Enabling or Disabling a Layer 7 Health Check” on page 12-31.

---

**NOTE:** DNS, HTTP, and RADIUS health checks use additional parameters, which you can configure using separate commands. See “Modifying the HTTP Keepalive Method, Value, and Status Codes” on page 12-32, “Configuring the DNS Health Check Method and Values” on page 12-38, or “Configuring the RADIUS Health Check Values” on page 12-38.

---

---

**NOTE:** When health checks are enabled for the ports on the VIPs in a host range, the ServerIron checks the health of the applications on the base IP address only. The ServerIron assumes that the health of an application is the same for all the VIPs within the host range. For information about host ranges, see “Web Hosting with Unlimited Virtual IP Addresses” on page 6-101.

---

## USING THE CLI

### Adding a Port and Specifying the Type

To add a port and specify that it is a TCP port, enter the following command:

```
ServerIron(config)# server port 8080
ServerIron(config-port-8080)# tcp
```

**Syntax:** server port <TCP/UDP-portnum>

**Syntax:** tcp | udp [keepalive [disable | enable]]

By adding a port, you also automatically enable periodic Layer 4 (and Layer 7, if applicable) keepalive health checks for the port. If you do not specify the port type (TCP or UDP), the ServerIron assumes that the port type is UDP.

### Changing a Port's Keepalive Parameters

To change a port's keepalive state, enter a command such as the following:

```
ServerIron(config-port-8080)# tcp keepalive disable
```

To change a port's keepalive interval and retries, enter a command such as the following:

```
ServerIron(config-port-80)# tcp keepalive 15 5
```

**Syntax:** tcp | udp keepalive [<interval> <retries>]

You can specify from 2 – 120 seconds for the interval. You can specify from 1 – 5 retries.

### USING THE WEB MANAGEMENT INTERFACE

To add a port profile and globally enable a health check, use the following procedure:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the [TCP/UDP Port](#) link to display the TCP/UDP Port Profile panel.
5. Select the [Add TCP/UDP Port](#) link to display a panel such as the one shown in Figure 12.1.

**Figure 12.1 TCP/UDP Port Profile panel**

The screenshot shows the 'Foundry Networks Device Management - Microsoft Internet Explorer' window. The address bar shows 'http://209.157.20.2/'. The left navigation pane is expanded to 'SLB' and then 'TCP/UDP Port'. The main content area displays the 'TCP/UDP Port Profile' configuration form. The form has a 'TCP/UDP Port' dropdown menu set to 'FTP' and a 'User Define' button. Below this are two sections: 'TCP' and 'UDP'. Each section has a checkbox for 'Keep Alive' (both checked), a text input for 'Age' (both set to 0), a text input for 'Keep Alive Interval' (both set to 5), and a text input for 'Keep Alive Retries' (both set to 2). At the bottom of the form are four buttons: 'Add', 'Modify', 'Delete', and 'Reset'. Below the buttons is a link 'Show TCP/UDP Port'. At the very bottom of the page are links: '[Home][Site Map][Logout][Save][Frame Enable/Disable][TELNET]'. The status bar at the bottom shows 'Internet'.

6. Select the port from the TCP/UDP Port field's pulldown list. If you are adding a new port that does not appear in the list, select User Define, then enter the port number in the TCP/UDP port field.
7. Select the port type by selecting the box next to TCP or UDP to place a checkmark in the box.
8. Change port parameters if desired. Notice that the health check (keep alive) is enabled by default.
9. Select Add if you are adding a new port profile or select Modify if you are changing a profile that was already configured.
10. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### ***Changing a Port's Session Age***

To change the age of session table entries for a port, enter a command such as the following:

```
ServerIron(config-port-80)# tcp 15
```

**Syntax:** server port <TCP/UDP-portnum>

**Syntax:** tcp | udp <2-60>

You can specify from 2 – 60 minutes.

### ***Basing a Port's Health on the Health of Another Port***

You can configure the ServerIron to base the health of a port that is not well-known to the ServerIron on the health of one of the following ports that are well-known to the ServerIron:

- DNS – the well-known name for port 53
- FTP – the well-known name for port 21. (Ports 20 and 21 both are FTP ports but on the ServerIron, the name "FTP" corresponds to port 21.)
- HTTP – the well-known name for port 80
- IMAP4 – the well-known name for port 143
- LDAP – the well-known name for port 389
- POP3 – the well-known name for port 110
- NNTP – the well-known name for port 119
- SMTP – the well-known name for port 25
- TELNET – the well-known name for port 23

To base a port's health on the health of another port, enter a command such as the following:

```
ServerIron(config-port-1234)# tcp keepalive port 80
```

**Syntax:** tcp | udp keepalive port <TCP/UDP-portnum>

The command in this example configures the ServerIron to base the health of port 1234 on the health of port 80 (HTTP). If the health of port 80 changes, the ServerIron applies the change to port 1234.

---

**NOTE:** You cannot base the health of a port well-known to the ServerIron on the health of another port, whether the port is well-known or not well-known.

---

### ***Basing an Alias Port's Health on the Health of its Master Port***

By default, the ServerIron performs health checks for alias ports independently of the master ports on which they are based. For example, if you configure alias port 8080 and base the port on port 80 (its master port), the ServerIron checks the health of 80 and 8080 independently.

You can configure the ServerIron to check the health of the master port only, and base the health of the alias ports on the master port.

You can base an alias port's health on the health of one of the following TCP ports:

- FTP – port 21 (ports 20 and 21 both are FTP ports but on the ServerIron, the name "FTP" corresponds to port 21)
- HTTP – port 80
- IMAP4 – port 143
- LDAP – port 389
- MMS – port 1755
- NNTP – port 119
- PNM – port 7070

- POP3 – port 110
- RTSP – port 554
- SMTP – port 25
- SSL – port 443
- TELNET – port 23

You cannot base an alias port's health on the health of a UDP port or a port that is not well-known to the ServerIron.

---

**NOTE:** The health checks for the alias ports must be enabled. Otherwise, the ServerIron will not check the master port's state, and the alias port will not go down when the master port goes down.

---

To configure an alias port's health to be based on its master port's health, edit the alias port's profile by entering commands such as the following:

```
ServerIron(config)# server port 8080
ServerIron(config-port-8080)# tcp keepalive use-master-state
```

**Syntax:** [no] tcp keepalive use-master-state

#### ***Overriding the Global TCP or UDP Age***

The TCP and UDP ages specify how many minutes a TCP or UDP session can remain inactive before the ServerIron closes the session and clears the session from its session table. You can set the TCP or UDP age from 2 – 60 minutes. The default TCP age is 30 minutes. The default UDP age is five minutes.

Since UDP is a connectionless protocol, the ServerIron does not remove a UDP session from its session table until the session times out. TCP is a connection-based protocol. Thus, for TCP sessions, the ServerIron removes the session as soon as the client or server closes the session.

---

**NOTE:** The ServerIron immediately deletes a UDP DNS or RADIUS session table entry when the ServerIron receives a reply for the application from a real server. If desired, you can configure the ServerIron to age these ports like other UDP ports, using the UDP age timer. See "Normal UDP Aging for DNS and RADIUS" on page 6-60.

---

To change the global default for all TCP or UDP ports, see "Modifying the TCP Age" on page 12-59 or "Modifying the UDP Age" on page 12-59.

To override the default for a specific TCP or UDP port, use one of the following methods.

#### ***USING THE CLI***

To override the default TCP age and set the age for TCP port 80 to 15 minutes, enter the following commands:

```
ServerIron(config)# server port 80
ServerIron(config-port-80)# tcp 15
```

**Syntax:** server port <TCP/UDP-portnum>

**Syntax:** tcp | udp <2-60>

You can specify from 2 – 60 minutes.

#### ***USING THE WEB MANAGEMENT INTERFACE***

To override the default TCP or UDP age for a specific port:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the TCP/UDP Port link to display the TCP/UDP Port Profile panel.

5. Click the Modify button next to the port to be modified. A panel such as the one shown in Figure 12.1 is displayed.
6. Edit the value in the Age field.
7. Click the Modify button.
8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### **Enabling Session Synchronization**

In Symmetric SLB configurations, if the active ServerIron becomes unavailable, service for the VIPs that ServerIron was load balancing is assumed by the backup ServerIron. By default, open sessions on the ServerIron that becomes unavailable are not carried over to the standby ServerIron. Instead, the sessions end and must be re-established by the clients or servers.

You can configure session failover on an individual TCP or UDP port basis by enabling session synchronization \in the port's profile.

### **USING THE CLI**

To enable session synchronization for port 80, enter the following commands:

```
ServerIron(config)# server port 80
ServerIron(config-port-80)# session-sync
```

**Syntax:** [no] server port <tcp/udp-portnum>

**Syntax:** [no] session-sync

### **USING THE WEB MANAGEMENT INTERFACE**

You cannot enable session synchronization using the Web management interface.

### **Changing the Smooth Factor on an Application Port**

This smooth factor applies to ports that you plan to use with the server response time load-balancing metric. See "Load Balancing Method (Predictor)" on page 6-24 and "Smooth Factor" on page 6-62 for information about the server response time metric and how the smooth time works.

The ServerIron calculates the server response time value for a real server by regularly collecting response time samples, then using a calculation to smooth the values of the samples and derive a single response time value for the real server. The ServerIron collects the samples around once every 100 milliseconds (about 10 times a second). The sampling rate can vary slightly depending on the processing the ServerIron is performing.

### **USING THE CLI**

To change the smooth factor for an application port, enter a command such as the following:

```
ServerIron(config-port-80)# smooth-factor 50
```

**Syntax:** smooth-factor <num>

### **USING THE WEB MANAGEMENT INTERFACE**

You cannot configure this parameter using the Web management interface.

### **Recursive Lookups for DNS Health Checks**

By default, a Layer 7 health check for a DNS port sends the query only to the real server (DNS server). If the DNS server does not reply with the IP address or zone name requested by the health check, the port fails the health check.

You can enable the real server to perform a recursive lookup for the IP address or zone requested by the health check. In this case, if the real server does not have the requested address or zone, the server can pass the request on to a DNS server with higher authority. The real server can repeat this process until either a DNS server with higher authority successfully replies to the health check or the server with the highest authority is unable to successfully reply to the request.



**NOTE:** Recursive DNS health checks are supported only on a ServerIron 400 or ServerIron 800 running software release 07.2.25 or later.

You can enable recursive DNS health checks globally at the port profile level for the DNS port.

```
ServerIron(config)# server port dns
ServerIron(config-port-dns)# allow-recursive-search
```

**Syntax:** [no] allow-recursive-search

**NOTE:** You can enable this feature only on the well-known DNS port (53).

---

### Modifying the Reassign Threshold

In addition to the circumstances described in “Server and Application Port States” on page 12-15, a real server’s state also can be affected by the reassign threshold. The **reassign threshold** specifies how many consecutive TCP SYN requests a real server can fail to respond to before the ServerIron changes the application state to FAILED and the server state to test. The default reassign threshold is 21.

The value of an application’s Reas field is reset to 0 when the ServerIron receives a TCP SYN ACK from the application. No other type of traffic can clear this field.

**NOTE:** The reassign threshold does not apply to servers in SwitchBack (Direct Server Return) configurations. In a SwitchBack configuration, traffic from the real server does not pass back through the ServerIron. As a result, the ServerIron cannot monitor the TCP SYN ACKs from the server. See “Configuring Symmetric SLB and SwitchBack” on page 7-1.

**NOTE:** The ServerIron does not try to reassign the client’s request to another server if you configure the application port to be sticky. The sticky option configures the ServerIron to override load-balancing and send all client requests for the application to the same server during a given session.

**NOTE:** If a real server seems to be triggering the reassign threshold too frequently, you can increase the reassign threshold. The default is 21 and the range of values is 6 – 254. This is a global parameter.

---

### USING THE CLI

To modify the SYN-ACK threshold to 215:

```
ServerIron(config)# server reassign-threshold 215
```

**Syntax:** server reassign-threshold <6-254>

---

### USING THE WEB MANAGEMENT INTERFACE

To modify the SYN-ACK threshold parameter:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu. The panel shown in Figure 6.8 on page 6-22 will appear.
5. Enter a value from 6 – 254 in the Reassign Threshold field.
6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device’s flash memory.

---

### Disabling the Reassignment Counter

By default, the ServerIron brings an application port down if the port’s reassignment count exceeds the reassign threshold. The default reassign threshold is 21. If a port fails to respond with a SYN ACK to 21 client SYNs in a

row, the ServerIron marks the port failed. Once the port is marked failed, the port can be re-activated as a result of a successful health check on the port.

In some networks, the reassignment counter can cause needless state flapping of application ports. This occurs if the network conditions cause the counter to frequently reach the threshold and cause the ServerIron to bring ports down even though the applications are healthy. The applications will remain unavailable for the amount of time it takes the ServerIron to send health checks, interpret the results, and activate the ports in response to successful results.

---

**NOTE:** The reassignment count applies to the total number of contiguous (back-to-back) unanswered SYNs from all clients who have sent SYNs to the server.

---

To prevent state flapping caused by the reassignment counter, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# server no-reassign-count
```

**Syntax:** [no] server no-reassign-count

When you enter this command, the ServerIron will stop incrementing the reassignment counters for real server applications.

---

**NOTE:** This command is supported only on the ServerIron 400 and ServerIron 800.

---

### Enabling the Health-Checking Procedure Used in Releases Prior to 7.1.05

In release 07.1.05, the health-checking procedure for application ports changed as follows:

- In releases prior to 07.1.05, the ServerIron performed a Layer 4 health check on a port on a real server, followed by a Layer 7 health check, if one was enabled on the port. If the port passed both health checks, it was then marked ACTIVE.
- Starting with release 07.1.05, by default when a port passes a Layer 4 health check, it is then marked ACTIVE. The ServerIron then performs a Layer 7 health check, if one is enabled on the port. Based on the result of the Layer 7 health check (if enabled), the port is then marked ACTIVE or FAILED.

This change was made so that ports could be brought up more quickly. You can optionally change the default behavior so that a port is not marked ACTIVE until it passes both the Layer 4 and (if one is enabled) Layer 7 health checks. In other words, you can optionally use the health-checking procedure that existed in releases prior to 07.1.05. To do so, enter the following command:

```
ServerIron(config)# server no-fast-bringup
```

**Syntax:** [no] server no-fast-bringup

### Using Simple SSL Health Checks (ServerIronXL Only)

In release 07.1.18, the SSL health checking procedure on the ServerIronXL changed. In this new procedure, the ServerIron negotiated an SSL connection and sent a GET or HEAD request to the server. This change applied to the ServerIronXL only. This change did not apply to other ServerIron models.

In previous releases, the SSL health check worked by sending the server an SSL client hello with the SSL SID set to 0. If the server responded, then the server passed the health check. The ServerIron then reset the connection and marked the SSL port ACTIVE. In this release, you can configure the ServerIronXL to use the pre-07.1.18 SSL health check procedure instead of the new one. To do this, enter the following command:

```
ServerIron(config)# server use-simple-ssl-health-check
```

**Syntax:** [no] server use-simple-ssl-health-check

For complete descriptions of both SSL health check methods, see “Health Check Summary” on page 12-4.

## Customizing Layer 7 Health Checks

You can configure the following Layer 7 health check parameters on a real server basis:

- Keepalive health check state (enabled or disabled)
- HTTP keepalive method, values, and valid status codes
- HTTP content matching lists for HTTP content verification health checks
- Scripted health checks (content verification health checks for unknown ports)
- DNS keepalive method and values (zone-based or addressed-based check and the zone or domain name)
- RADIUS keepalive values (user name, password, and encryption key)
- LDAP version (2 or 3)

---

**NOTE:** The ServerIron uses its own management IP address or a source IP address configured on the ServerIron as the source IP address in the health check packets (as opposed to a virtual IP address). If the real servers are in the same sub-net as the ServerIron, then the health checks can use the ServerIron's management IP address. Otherwise, the health checks use a source IP address. See "Web Hosting with ServerIron and Real Servers in Different Sub-Nets" on page 6-107.

---

## Locally Enabling or Disabling a Layer 7 Health Check

All Layer 7 health checks are disabled by default. You can enable a health check globally or locally. To enable or disable a health check locally, use one of the following methods.

---

**NOTE:** The ServerIron considers a Layer 7 health check to be disabled only if the health check is disabled on both the global and local levels. If the health check is enabled globally, locally, or both globally and locally, the ServerIron considers the health check to be enabled. See "Adding a TCP or UDP Port, Specifying the Port Type, and Configuring the Keepalive Health Check" on page 12-23.

---

### USING THE CLI

To locally enable a Layer 7 health check, enter a command such as the following at the Real Server level of the CLI:

```
ServerIron(config-rs-jet)# port dns keepalive
```

**Syntax:** [no] port <port> keepalive

If you use the "no" parameter in front of the command, you are locally disabling the health check. The health checks are locally disabled by default.

The <port> parameter can have one of the following values:

- **dns** – the well-known name for port 53
- **ftp** – the well-known name for port 21. (Ports 20 and 21 both are FTP ports but in the ServerIron, the name "ftp" corresponds to port 21.)
- **http** – the well-known name for port 80
- **imap4** – the well-known name for port 143
- **ldap** – the well-known name for port 389
- **nntp** – the well-known name for port 119
- **ntp** – the well-known name for port 123
- **pop2** – the well-known name for port 109
- **pop3** – the well-known name for port 110
- **radius** – the well-known name for UDP port 1812

- **radius-old** – the ServerIron name for UDP port 1645, which is used in some older RADIUS implementations instead of port 1812
- **smtp** – the well-known name for port 25
- **snmp** – the well-known name for port 161
- **ssl** – the well-known name for port 443
- **telnet** – the well-known name for port 23
- **tftp** – the well-known name for port 69
- <number>

---

**NOTE:** Specify the port number if the port is not one of the well-known names listed above.

---

#### USING THE WEB MANAGEMENT INTERFACE

This parameter cannot be configured using the Web management interface.

#### Modifying the HTTP Keepalive Method, Value, and Status Codes

The ServerIron supports two kinds of HTTP health checks:

- **HTTP status code** health checks look at the status code returned in HTTP responses to keepalive requests.
- **HTTP content verification** health checks look at the actual HTML contained in HTTP responses to keepalive requests.

The default URL page for HTTP keepalive requests used in HTTP health checks is “HEAD /1.0”. You can change the URL that the ServerIron requests on a real server basis.

---

**NOTE:** For HTTP content verification health checks, you may want to change the default URL page for HTTP keepalive requests URL page, since a request for “HEAD /1.0” would not return a response containing HTML for content verification. You can specify a GET request for a page containing text that can be searched and verified. See “Configuring HTTP Content Matching Lists” on page 12-33 for more information.

---

#### USING THE CLI

To configure the HTTP keepalive request to send a GET request for “sales.html”, enter the following commands:

```
ServerIron(config)# server real zip 207.96.3.251
ServerIron(config-rs-zip)# port http url "GET/sales.html"
ServerIron(config-rs-zip)# exit
```

```
ServerIron(config)# server virtual shoosh 207.96.4.250
ServerIron(config-vs-shoosh)# port http
ServerIron(config-vs-shoosh)# bind http zip http
ServerIron(config-vs-shoosh)# exit
```

**Syntax:** port http url “[GET | HEAD] [/]<URL-page-name>”

GET or HEAD is an optional parameter that specifies the request type. By default, HTTP keepalive uses HEAD to retrieve the URL page. You can override the default and configure the ServerIron to use GET to retrieve the URL page.

The slash (/) is an optional parameter. If you do not set the GET or HEAD parameter, and the slash is not in the configured URL page, then ServerIron automatically inserts a slash before retrieving the URL page.

To change the HTTP status codes that the ServerIron considers normal (not indicative of a failure of the HTTP service), enter the following command.

```
ServerIron(config-rs-zip)# port http status-code 200 201 300 302
```

**Syntax:** port http status-code <range> [<range>[<range>[<range>]]]

The command in this example specifies two ranges (200 – 201 and 300 – 302). You can specify up to four ranges (total of eight values). To specify a single message code for a range, enter the code twice. For example to specify 200 only, enter the following command: **port http status-code 200 200**.

**NOTE:** When you change the status code ranges, the defaults are removed. As a result, you must specify all the valid ranges, even if a range also is within the default ranges. For example, if you still want codes 200 – 299 to be valid, you must specify them. For the defaults, see “HTTP Status Codes” on page C-1.

### USING THE WEB MANAGEMENT INTERFACE

1. Select the [Real Server Port](#) link from the bottom of the General SLB panel or another SLB panel.
2. Select the Modify button next to the real server to be modified. The panel shown in Figure 12.2 will appear.

**Figure 12.2** Real server port entry panel

Real Server Port	
Server Name:	test
TCP/UDP Port:	HTTP <input type="button" value="User Define"/>
Status:	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Keep Alive:	<input type="checkbox"/>
DNS Parameters	
+DNS Zone:	
+Addr Query:	
+Proxy:	<input type="checkbox"/>
HTTP Parameters	
*Method:	HEAD
*URL:	
*Status Code:	
Group Id Range	
From	To
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

**NOTE:** Some of the fields on the Real Server Port panel apply only to HTTP while other fields apply only to DNS.

3. Select the method (HEAD or GET) from the Method field.
4. Enter the URL that the HTTP health check will request in the URL field.
5. Select the Modify button to implement the change.
6. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Configuring HTTP Content Matching Lists

An **HTTP content verification** health check is a type of Layer 7 health check in which the ServerIron examines text in an HTML file sent by a real server in response to an HTTP keepalive request. The ServerIron searches the text in the HTML file for user-specified selection criteria and determines whether the HTTP port on the real server is alive based on what it finds.

The selection criteria used in HTTP content verification is contained in a **matching list** that is bound to one or more real servers. The following is an example of the commands used to set up a matching list.

```
ServerIron(config)# http match-list m1
ServerIron(config-http-m1-m1)# down simple "404"
ServerIron(config-http-m1-m1)# down simple "File Not Found"
ServerIron(config-http-m1-m1)# exit
```

**Syntax:** http match-list <matching-list-name>

**Syntax:** down simple <text> [log]

The **http match-list m1** command sets the name of the matching list and enters the HTTP matching list CLI level.

The **down simple** statements specify the selection criteria in this matching list. In the example above, the first statement looks for the text "404" in the HTML file sent from the real server in response to an HTTP keepalive request; the second statement looks for the text "File Not Found". If either of these text strings are found in the HTML file, the ServerIron marks port 80 (HTTP) on the real server FAILED. If neither of the text strings are found, the ServerIron marks the port ACTIVE.

---

**NOTE:** There is a limit of 200 selection criteria statements for all HTTP matching lists. That is, the total number of **up** and **down** statements in all HTTP matching lists on the ServerIron must not exceed 200.

---

When an HTML file meets more than one set of selection criteria in a matching list, the ServerIron takes one of the following actions:

- If the strings that meet the selection criteria are different, the ServerIron takes action based on the string that comes first in the file. For example:

```
ServerIron(config)# http match-list m2
ServerIron(config-http-m1-m2)# down simple "monkey"
ServerIron(config-http-m1-m2)# up simple "elephant"
ServerIron(config-http-m1-m2)# exit
```

The selection criteria in the matching list above would cause the ServerIron to mark the port FAILED if the text "monkey" is found and ACTIVE if the text "elephant" is found. If the HTML file has the text "monkey" at the beginning and "elephant" at the end, the ServerIron would mark port 80 on the real server FAILED, because "monkey" occurs first in the file.

- If a string that meets the selection criteria is a subset of another, the longer string takes precedence, regardless of where it occurs in the file. For example:

```
ServerIron(config)# http match-list m3
ServerIron(config-http-m1-m3)# down simple "elephant"
ServerIron(config-http-m1-m3)# up simple "elephantine"
ServerIron(config-http-m1-m3)# exit
```

In this example, ServerIron would mark the port FAILED if the text "elephant" is found and ACTIVE if the text "elephantine" is found. If the HTML file has the text "elephant" at the beginning and "elephantine" at the end, the ServerIron would mark port 80 on the real server ACTIVE, because "elephantine" is longer than "elephant".

The following is an example of a matching list that uses **compound** selection criteria, in which the beginning and ending parts of selection criteria are specified:

```
ServerIron(config)# http match-list m4
ServerIron(config-http-m1-m4)# up compound "monkey see" "monkey do" log
ServerIron(config-http-m1-m4)# down compound "500" "Internal Server Error" log
ServerIron(config-http-m1-m4)# down compound "503" "Service Unavailable" log
ServerIron(config-http-m1-m4)# default down
ServerIron(config-http-m1-m4)# exit
```

**Syntax:** up compound <start> <end> [log]

**Syntax:** down compound <start> <end> [log]

**Syntax:** default down | up

In this matching list, the **up** and **down** commands include the **compound** parameter, which allows you to specify beginning and ending parts of a set of selection criteria. Text that begins with the first part and ends with the second part meets the selection criteria.

In this example, the **up** command specifies that if the HTML file sent from the real server in response to an HTTP keepalive request contains a text string that begins with the text "monkey see" and ends with the text "monkey do", port 80 on the real server is marked ACTIVE. The **down** commands specify that if the HTML file contains a text string that begins with "500" and ends with "Internal Server Error" or begins with "503" and ends with "Service Unavailable", the port is marked FAILED.

The **default** command specifies what happens if none of the HTML text in the HTTP response message meets the selection criteria. You can specify either **up** or **down**; the default is **up**. In this example, the **default down** command causes port 80 on the real server to be marked FAILED if none of the selection criteria are found in the HTTP response message.

The commands in the matching list also include the **log** parameter, which causes the following Warning message to be logged when the selection criteria is met:

```
00d00h00m00s:W:HTTP match-list <matching-list> with compound pattern1 <start> and pattern2 <end> Alert:
bring server down and Extract message: <text-between-start-and-end-pattern>
```

In the example above, the following message would be logged at the successful completion of an HTTP content verification health check; that is, if the HTML file sent from the real server in response to an HTTP keepalive request contains a text string that begins with the text "monkey see" and ends with the text "monkey do":

```
ServerIron# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Buffer logging: level ACDMEINW, 1 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                 I=informational N=notification W=warning

Dynamic Log Buffer (50 entries):
02d04h47m12s:W:HTTP match-list m4 with compound pattern1 "monkey see" and pattern2
"monkey do" Alert: bring server up and Extract message: This web page is configured
correctly
```

### Displaying Matching List Information

To display the contents of matching lists configured on the ServerIron, enter the following command:

```
ServerIron# show http match-list
http match-list m1
  down simple "404"
  down simple "File Not Found"
http match-list m4
  default down
  up compound "monkey see" "monkey do" log
  down compound "500" "Internal Server Error" log
  down compound "503" "Service Unavailable" log
```

### Binding the Matching list to the Real Servers

To enable HTTP content verification on the ServerIron, you bind the matching list to one or more real servers. For example:

```
ServerIron(config)# server real-name rs1 192.168.1.1
ServerIron(config-rs-rs1)# port http content-match m4
ServerIron(config-rs-rs1)# port http url "GET/monkey.html"
ServerIron(config-rs-rs1)# exit
```

**Syntax:** server real-name <real-server-name> <ip-addr>

**Syntax:** port http content-match <matching-list-name>

**Syntax:** port http url "[GET | HEAD] [/]<URL-page-name>"

In this example, the **port http content-match m4** command binds matching list m4 to real server rs1. HTTP response messages coming from real server rs1 are examined using the selection criteria in matching list m4.

The **port http url** command sets the method used for HTTP keepalive requests and the URL of the page to be retrieved. This command is used in HTTP content verification health checks because the default method and URL page for HTTP keepalive requests used in HTTP health checks, "HEAD /1.0", does not return an HTML file that the ServerIron can search and verify. You can instead specify the GET method, which does return an HTML file that can be examined using the matching list.

### Configuring Scripted Health Checks

You can configure **scripted health checks**, which are content verification health checks for ports that do not use one of the well-known port numbers recognized by the ServerIron.

In a scripted health check, the ServerIron opens a connection to a port on a real server by sending a SYN packet. The ServerIron then waits for the real server to send back a packet in response. The ServerIron looks in the response packet for a user-specified ASCII string, defined in a matching list on the ServerIron. The port on the real server is then marked ACTIVE or FAILED, based on configuration settings in the matching list. For example, a matching list can be configured to mark a port ACTIVE or FAILED if the string is found, or mark the port ACTIVE or FAILED if the string is not found.

If no response is received within the configured interval (the default is five seconds), the ServerIron sends a RST and retries the health check. After the configured number of retries (the default is two retries), if the server still does not respond, the ServerIron marks the server port FAILED.

A scripted health check can also be part of a health-check policy. In this case, the scripted health check checks the health of a configured port in the policy. The health-check policy can be evaluated to true or false depending on the response from the server.

Setting up a scripted health check consists of the following steps:

- Creating a port profile
- Creating a matching list
- Binding the matching list to the real server

#### Creating a Port Profile

A scripted health check will not work on a TCP port that does not have a profile associated with it, since the ServerIron assumes any port without a profile is a UDP port, and will perform UDP health checking on the port. To use a scripted health check on a TCP port, you must create a port profile and explicitly identify the port as a TCP port.

The following commands configure a port profile for port 12345 and specify that the port is a TCP port. The **no-fast-bringup** command is necessary because it prevents the ServerIron from marking a port ACTIVE until it passes both Layer 4 and Layer 7 health checks.

```
ServerIron(config)# server port 12345
ServerIron(config-port-12345)# tcp
ServerIron(config-port-12345)# no-fast-bringup
```

**Syntax:** server port <TCP/UDP-portnum>

**Syntax:** tcp | udp [keepalive <interval> <retries>]

**Syntax:** no-fast-bringup

#### Creating a Matching List

The selection criteria used in a content verification health check is specified in a matching list that is bound to one or more real servers. The syntax used for creating a matching list for scripted health checks is the same as that used for a creating matching list for HTTP content verification health checks in "Configuring HTTP Content Matching Lists" on page 12-33.



The following is an example of a matching list that will mark a port ACTIVE if the string "FTP service" is found in the response from the real server. If this text is not found, the port on the real server is marked FAILED.

```
ServerIron(config)# http match-list m1
ServerIron(config-http-m1-m1)# up simple "FTP service"
ServerIron(config-http-m1-m1)# default down
ServerIron(config-http-m1-m1)# exit
```

**Syntax:** http match-list <matching-list-name>

**Syntax:** up simple <text> [log]

**Syntax:** default down | up

The **up simple** statement specifies the selection criteria in this matching list.

The **default** command specifies what happens if none of the text in the response from the real server meets the selection criteria. You can specify either **up** or **down**; the default is **up**. In this example, the **default down** command causes the port on the real server to be marked FAILED if the selection criteria is not found in the response from the server. If the real server responds to the health check with a RST, the port is marked ACTIVE or FAILED depending on what was specified in the **default** statement in the matching list.

You can also use the **compound** parameter, which allows you to specify beginning and ending parts of a set of selection criteria, as well as the **log** parameter, which causes the a Warning message to be logged when the selection criteria is met.

#### ***Binding the Matching List to the Real Server***

To enable the scripted health check on the ServerIron, you bind the matching list to one or more real servers. For example, to bind matching list m1 to real server R:

```
ServerIron(config)# server real R 10.10.10.50
ServerIron(config-rs-R)# port 12345 content-check m1
```

**Syntax:** port <portnum> content-check <matching-list-name>

The <portnum> is a non-well-known port. You cannot specify a well-known port for a scripted health check.

The <matching-list-name> is a previously configured matching list. If the <matching-list-name> does not refer to an existing matching list, the port on the real server is marked FAILED when the health check is performed.

#### ***Using a Scripted Health Check in a Health-Check Policy***

A scripted health check can be used in a health-check policy. A health-check policy is a group of one or more health checks attached to a real server port. When the scripted health check checks the health of a destination port specified in the policy, the health-check policy can be evaluated to true or false depending on the response from the server.

To use a scripted health check with a health-check policy, you configure a matching list, then configure the health-check policy.

For example, when the following matching list is used with a health-check policy, it will evaluate the policy to true if the string "FTP service" is found in the response from the real server. If this text is not found, the policy is evaluated to false.

```
ServerIron(config)# http match-list m1
ServerIron(config-http-m1-m1)# up simple "FTP service"
ServerIron(config-http-m1-m1)# default down
ServerIron(config-http-m1-m1)# exit
```

The **default down** command causes the policy to be evaluated to false if the selection criteria is not found in the response from the server. If the real server responds to the health check with a RST, the policy is evaluated to true or false depending on what was specified in the **default** statement in the matching list.

The following commands create a health check policy for TCP port 1234 on VIP 10.10.10.10. Matching list m1 is bound to this policy.

```
ServerIron(config)# healthck check1 tcp
ServerIron(config-hc-check1)# dest-ip 10.10.10.10
```

```
ServerIron(config-hc-check1)# port 1234 content-check m1
ServerIron(config-hc-check1)# l7-check
```

**Syntax:** [no] healthck <element-name> <protocol>

**Syntax:** [no] dest-ip <ip-addr>

**Syntax:** [no] port <portnum> content-check <matching-list-name>

**Syntax:** [no] l7-check

Note that the **dest-ip** <ip-addr> command **must** be the first command entered for a health-check policy. If this is not the first command entered for the policy, an error message is displayed.

If the <matching-list-name> does not refer to an existing matching list, the policy is evaluated to false.

The **l7-check** command is required to ensure that the ServerIron performs a Layer 7 health check. If this command is omitted, the ServerIron performs only a Layer 4 health check, and not the scripted health check.

### Configuring the DNS Health Check Method and Values

The keepalive time and number of retries are global parameters. However, you configure the DNS health checking methods and values on an individual server basis. You can configure the following types of DNS health checks:

- Address-based – The ServerIron sends an address request for a specific domain name. If the server successfully responds with the IP address for the domain name, the server passes the health check.
- Zone-based – The ServerIron sends a Source-of-Authority (SOA) request for a specific zone name. If the server is authoritative for the zone and successfully responds to the SOA request, the server passes the health check.

#### USING THE CLI

To configure the domain name for address-based DNS health checking, enter the following command:

```
ServerIron(config-rs-zip)# port dns addr_query "evil.mojo.com"
```

**Syntax:** [no] port dns addr\_query "<name>"

To configure the zone name for zone-based DNS health checking, enter the following command:

```
ServerIron(config-rs-zip)# port dns zone mojo.com
```

**Syntax:** [no] port dns zone <zone-name>

#### USING THE WEB MANAGEMENT INTERFACE

1. Select the [Real Server Port](#) link from the bottom of the General SLB panel or another SLB panel.
2. Select the Modify button next to the real server to be modified. The panel shown in Figure 12.2 will appear.

---

**NOTE:** Some of the fields on the Real Server Port panel apply only to HTTP while other fields apply only to DNS.

---

3. Select the name of the real server from the Server Name field's pulldown menu.
4. Enter the zone name in the DNS Zone field.
5. Select the Modify button to implement the change.
6. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

### Configuring the RADIUS Health Check Values

The RADIUS health check requests a specific user name, password, and authentication key from the RADIUS server. To specify these values, use one of the following methods.

### USING THE CLI

To configure the parameters for a RADIUS health check, enter commands such as the following at the Real Server level of the CLI:

```
ServerIron(config-rs-rocket)# port radius username evil
ServerIron(config-rs-rocket)# port radius password woody
ServerIron(config-rs-rocket)# port radius key laser
```

**Syntax:** [no] port radius username <string>

**Syntax:** [no] port radius password <string>

**Syntax:** [no] port radius key <string>

### USING THE WEB MANAGEMENT INTERFACE

You cannot configure the RADIUS health check parameters using the Web management interface.

### Changing the LDAP Version

By default, the ServerIron Layer 7 health check for LDAP ports tests for version 3 LDAP. You can change the version to 2 if needed. To do so, use either of the following methods.

#### USING THE CLI

To change the LDAP version the ServerIron uses when checking the health of an LDAP port on a real server, enter a command such as the following at the Real Server level of the CLI:

```
ServerIron(config-rs-rocket)# port ldap 2
```

**Syntax:** [no] port ldap <num>

The <num> parameter specifies the version and can be 2 or 3. The default is 3.

#### USING THE WEB MANAGEMENT INTERFACE

You cannot configure LDAP health check parameters using the Web management interface.

### Checking the Health of Multiple Web Sites on the Same Real Server

If you host multiple web sites on the same real server, with each web site using a different VIP, you can perform an independent health check for each VIP.

As described in “Many-To-One TCP/UDP Port Binding” on page 6-98, to bind two VIPs to the HTTP port on the same real server, you create an alias for the HTTP port on one of the VIPs. To create an alias for the HTTP port, you configure the VIP to bind to an alternate port number on the real server, then disable port translation for that binding. The ServerIron collects and presents information for the alias port number, but traffic from both VIPs actually goes to the HTTP port on the real server.

The state of the master port is used to indicate the health of ports aliased to the master port. For example, if a VIP uses port 81 as an alias for the HTTP port, then the state information reported for the HTTP port is used as the state information for port 81. If the HTTP port is reported down, then port 81 is reported down.

When a real server supports multiple web sites, tying the alias port's state to the master port's state may cause incorrect information to be reported. For example, consider a real server hosting VIPs v1 and v2. VIP v1 is bound to the HTTP port on the real server, and VIP v2 uses port 81 as an alias for the HTTP port. The Layer 7 health check reports state information about the HTTP port. When VIP v1 is taken down for maintenance, the Layer 7 health check reports that the HTTP port is down. Because the state information reported for the HTTP port is also used as the state information for port 81, the ServerIron considers port 81 to be down as well, incorrectly reflecting the state of VIP v2, which may be functioning normally.

To eliminate this problem, you establish separate health checks for the alias ports. Health checks for the alias ports will continue to be performed regardless of the HTTP port's status. The following is an example of this kind of configuration:

```
ServerIron(config)# server virtual v1 192.168.1.160
ServerIron(config-vs-v1)# port http
ServerIron(config-vs-v1)# bind http rs32 http
```

```
ServerIron(config-vs-v1)# exit

ServerIron(config)# server virtual v2 192.168.1.161
ServerIron(config-vs-v2)# port http
ServerIron(config-vs-v2)# no port http translate
ServerIron(config-vs-v2)# bind http rs32 81
ServerIron(config-vs-v2)# exit

ServerIron(config)# server real rs32 64.1.1.32
ServerIron(config-rs-rs32)# port http
ServerIron(config-rs-rs32)# port http keepalive
ServerIron(config-rs-rs32)# port http url "HEAD /"
ServerIron(config-rs-rs32)# port 81
ServerIron(config-rs-rs32)# port 81 keepalive
ServerIron(config-rs-rs32)# port 81 url "GET /81keepalive.htm"
ServerIron(config-rs-rs32)# exit
```

In this configuration, two VIPs are bound to a single real server. VIP v2 uses port 81 as an alias for port 80; information the ServerIron receives about port 81 is attributed to VIP v2. If VIP v1 is taken down for maintenance, the Layer 7 health check done for port 80 fails, and the ServerIron marks the HTTP port FAILED. However, health checks continue to be performed for port 81. Port 81 (and thus VIP v2) will continue to be reported active as long as it passes its health check.

## Using a Layer 7 Health Check for an Unknown Port

You can use Layer 7 health check parameters for the following known ports to check the health of unknown ports:

TCP ports:

- FTP (port 21)
- IMAP4 (port 143)
- LDAP (port 389)
- POP3 (port 110)
- SMTP (port 25)
- Telnet (port 23)

UDP ports:

- DNS (port 53)

## Using Layer 7 TCP Health Checks for Unknown Ports

---

**NOTE:** This feature is supported only in software release 07.2.16 and higher 07.2.x releases.

---

You can use the ServerIron's Layer 7 health check mechanism for the following TCP applications on any TCP port number:

- FTP (port 21)
- IMAP4 (port 143)
- LDAP (port 389)
- POP3 (port 110)
- SMTP (port 25)
- Telnet (port 23)

The health check mechanisms for these ports are described in "Health Check Summary" on page 12-4.

To configure an unknown TCP port to use the Layer 7 health check for one of the applications listed above, enter commands such as the following:

```
ServerIron(config)# server port 999
ServerIron(config-port-999)# tcp keepalive protocol smtp
```

These commands configure port profile parameters for port 999. The second command in the example makes the port a TCP port and assigns the SMTP Layer 7 health check to the port.

**Syntax:** [no] server port <TCP-portnum>

**Syntax:** [no] tcp keepalive protocol <TCP-port>

The **protocol** <TCP-port> parameter specifies the type of Layer 7 health you want to use for the port. You can specify one of the following:

- **ftp** or **21**
- **imap4** or **143**
- **ldap** or **389**
- **pop3** or **110**
- **smtp** or **25**
- **telnet** or **23**

### Configuring an Unknown UDP Port to Use a Layer 7 Health Check

The ServerIron can perform Layer 7 health checks on the DNS port (UDP port 53).

---

**NOTE:** This feature is supported only in software release 07.1.18 and higher 07.1.x releases.

---

To configure an unknown UDP port to use the DNS Layer 7 health check:

- Configure the Layer 7 health check on the DNS port (53). For configuration information, see “Configuring the DNS Health Check Method and Values” on page 12-38. The unknown port uses the same health check parameters as the ones you configure for the DNS port. For example, if you configure an address-based DNS health check for a specific domain name, the ServerIron requests the same domain name when checking the health of the unknown port.
- Create a port profile for the unknown port and specify **dns** or **53** as the well-known port whose Layer 7 health check you want to use.

To configure an unknown port to use a Layer 7 health check, enter commands such as the following:

```
ServerIron(config)# server port 999
ServerIron(config-port-999)# udp keepalive protocol dns
```

**Syntax:** server port <UDP-portnum>

**Syntax:** udp keepalive protocol <UDP-portnum>

The **protocol** <UDP-port> parameter specifies the type of Layer 7 health you want to use for the port. You can specify **dns** or **53**.

### Configuring Boolean Health-Check Policies (ServerIron 400 and ServerIron 800)

You can configure a group of Layer 4 and Layer 7 health checks as a health-check policy and associate the group with a specific application port on a real server.<sup>1</sup> Health-check policies enable you to assess the health of any application port using the health-check mechanisms for ports well-known to the ServerIron. In addition, health-check policies enable you to use multiple checks with different parameters, and base a port's health on successful completion of all or any one of the individual checks in the policy.

---

1. Real servers include those added using the **server real-name** command and those added using the **server remote-name** command. Generally, both types of servers are referred to as real servers. An application port is a port that uses the TCP or UDP protocol. You associate health-check policies with TCP or UDP ports on the real servers (not with physical ports on the servers).

**NOTE:** This section applies only to software release 07.2.23 and higher for the ServerIron 400 and ServerIron 800. To use Boolean health-check policies on the ServerIronXL, see “Configuring Boolean Health-Check Policies (ServerIronXL)” on page 12-52.

---

Depending on the conditions you specify when you configure a health-check policy, the ServerIron will bring the application port on a server down in one of the following cases:

- Any one of the servers fails its health check (individual health checks combined using AND condition) – In this case, all servers in the policy must pass their health checks. Otherwise, the ServerIron considers all of the servers to have failed the health checks and brings down the application on all servers that are checked by the policy.
- All of the servers fail their health checks (individual health checks combined using OR condition) – In this case, an application port remains up as long as at least one of the servers checked by the policy passes its health check.

For finer control, you can combine OR and AND conditions.

### Health-Check State

When you attach a health-check policy to a real server’s application port, the ServerIron uses the health-check policy for periodic health checks and also for the next initial bringup of the server. When a health-check policy is attached, the ServerIron no longer uses the default health check methods for initial bringup and periodic health checks described in “Health Check Summary” on page 12-4.

For the ServerIron to use a health-check policy, you must enable health checking (keepalive) at either the port profile level or the real server level for the server port. Otherwise, the state of the policy is FALSE and the state of the server port remains the state that it was before you attached the policy.

---

**NOTE:** Use the **show healthchk** command to display the policy state. Use the **show server real-name <name>** command to show the real server port state.

---

If health checking for a server port is disabled at the port profile level and also at the real server level, the ServerIron will continue to use the state that is based on the health check during the initial server bringup. The ServerIron will not be able to update the port’s state if the state changes.

To enable health checking at the port profile level, enter commands such as the following:

```
ServerIron(config)# server port 80
ServerIron(config-port-80)# tcp keepalive enable
```

The commands above enable health checking for TCP port 80. For a UDP port, enter commands such as the following:

```
ServerIron(config)# server port 53
ServerIron(config-port-53)# udp keepalive enable
```

To enable health checking at the real server level, enter commands such as the following:

```
ServerIron(config)# server real-name R1 10.10.10.10
ServerIron(config-rs-R1)# port 80 keepalive
```

You can enable health checking at the port profile level, at the real server level, or both. Health checking must be enabled on at least one of these levels for the ServerIron to use the health-check policy you attach to the port.

### Configuring a Health-Check Policy

Health-check policies consist of element-action expressions and logical expressions.

- Element-action expression – An element-action expression consists of the IP address of the server, the Layer 4 protocol (TCP or UDP), and the application port on the server. For some applications, the element-action expression can also include Layer 7 application-specific health check information.
- Logical expression – A logical expression is a set of element-action expressions joined by the Boolean operators OR and AND.

- To create a health-check policy that is successful if at least one of the applications passes its health check, use OR.
- To configure a health-check policy that is successful only if the ServerIron receives a successful reply from all servers and application ports in the policy, use the operator AND.

You can use the same element-action expressions in multiple logical expressions if desired. You can configure up to 254 health-check policies.

To use a health-check policy:

- Configure the element-action expressions.
- Configure the health-check policy using element-action expressions and logical expressions joined by the operators AND or OR.
- Attach logical expressions to application ports on specific real servers. A health check policy does not take effect until you attach it to an application port on a server.

---

**NOTE:** A health-check policy does not take effect (begin sending health check packets) until you attach the policy to an application port on a real server.

---

### ***Configuring an Element-Action Expression***

An element-action expression contains the IP address, protocol (TCP or UDP), and application port number for an application on an individual real server. If the ServerIron allows you to customize Layer 7 information for the application, then the element-action expression also can contain the customized Layer 7 information.

You also can change the following parameters for an application port when configuring an element-action expression:

- Health check type – For application types that are well-known to the ServerIron, you can specify whether you want to use the Layer 4 health check or the Layer 7 health check for the port. By default, the ServerIron uses the Layer 7 health check if the port is one of the types well-known to the ServerIron.
- Health check interval – By default, the ServerIron performs the health checks every 5 seconds. You can change the interval to a value from 2 – 120 seconds.
- Health retries – By default, if a reply to a health check is not received, the ServerIron will attempt the health check two more times before concluding that the application has failed the health check. You can change the number of retries to a value from 1 – 5 retries.
- Health check state – By default, the health check is enabled as soon as you configure it. You can disable or re-enable the health check from within the element-action expression for the check.

### **Specifying the IP Address and Application Port Parameters**

To configure an element-action expression, enter commands such as the following. The commands in this example specify the IP address of the real server and the application port on the server.

```
ServerIron(config)# healthck check1 tcp
ServerIron(config-hc-check1)# dest-ip 10.10.10.50
ServerIron(config-hc-check1)# port http
```

These commands change the CLI to the configuration level for an element-action expression, then specify the IP address of the real server and the application port on the server. Since the specified application is well-known to the ServerIron, the ServerIron automatically associates the default health check parameters for the port with the element-action expression. In this example, the port is HTTP (80), so the ServerIron associates the default HTTP health check parameters with the element-action expression. By default, the ServerIron sends a HEAD request for the default page, "1.0".

---

**NOTE:** You must specify the destination IP address before you can specify other health check parameters. The software creates the health check policy only after you specify the destination IP address. If you try to specify another parameter before the destination IP address, the CLI displays an error message such as the following: Error - check1: Health-check element is undefined.

---

**NOTE:** If you do not specify the application port, the ServerIron will list the status of the health check as FALSE (failed).

---

To configure an element-action expression for a port number that is not well-known to the ServerIron, enter commands such as the following:

```
ServerIron(config)# healthck check1 tcp
ServerIron(config-hc-check1)# dest-ip 10.10.10.50
ServerIron(config-hc-check1)# port 8080
ServerIron(config-hc-check1)# protocol http
```

These commands configure an element-action expression for unknown port 8080 and associate the default health check parameters for port 80 with the unknown port. To customize the Layer 7 health check parameters for a port, add the information with the **protocol** command, as in the following example:

```
ServerIron(config)# healthck check1 tcp
ServerIron(config-hc-check1)# dest-ip 10.10.10.50
ServerIron(config-hc-check1)# port 8080
ServerIron(config-hc-check1)# protocol http url "GET/sales.html"
```

The **protocol** command in this example changes the Layer 7 health check parameters for this HTTP port to a GET request for a page named "sales.html".

**Syntax:** [no] healthck <string> tcp | udp

This command begins configuration of the element-action expression. The <string> parameter specifies the name for the expression and can be up to 20 characters long. The **tcp | udp** parameter specifies whether you are configuring an expression for a TCP application port or a UDP application port. There is no default.

**Syntax:** [no] dest-ip <ip-addr>

This command specifies the IP address of the real server.

**Syntax:** [no] port <tcp/udp-port>

This command specifies the application port number.

---

**NOTE:** If you do not specify the server IP address and the application port, the ServerIron will list the status of the health check as FALSE (failed).

---

You can specify any valid number, or one of the following port names well-known to the ServerIron:

- **dns** – port 53
- **ftp** – port 21. (Ports 20 and 21 both are FTP ports but in the ServerIron, the name "ftp" corresponds to port 21.)
- **http** – port 80
- **imap4** – port 143
- **ldap** – port 389
- **nntp** – port 119
- **ntp** – port 123
- **pop2** – port 109
- **pop3** – port 110



- **radius** – port 1812
- **radius-old** – the ServerIron name for UDP port 1645, which is used in some older RADIUS implementations instead of port 1812
- **smtp** – port 25
- **snmp** – port 161
- **ssl** – port 443
- **telnet** – port 23
- **tftp** – port 69

---

**NOTE:** If you enter the **no port** <tcp/udp-port> command to remove the port, the ServerIron also removes the **protocol** <tcp/udp-port> command (see below) if the port is well-known to the ServerIron. This is because the ServerIron automatically uses the protocol that matches the well-known port. Otherwise, the ServerIron does not remove the protocol. You must remove it separately.

---

**Syntax:** [no] protocol <tcp/udp-port>

This command specifies a port whose health-check mechanism you want to use for the port specified by the **port** command. You need to use this command only if the port specified by the **port** command is not one of the ports listed above but the port is the same type as one of the ports listed above. For example, use this command if you want to use the DNS health-check mechanism for a port other than 53.

---

**NOTE:** You must specify the port using the **port** command before you enter the **protocol** command. If the **port** command specified a port that is well-known to the ServerIron, the ServerIron automatically uses the protocol that matches the port; you do not need to specify it and cannot change it.

---

**NOTE:** If you remove the Layer 7 health check information (using a **no protocol** command), the application will fail the health check. If you want the ServerIron to use a Layer 4 health check instead, enter the **l4-check** command to change the health-check type to Layer 4.

If the port is not well-known to the ServerIron and you do not specify a protocol for the Layer 7 health check, but Layer 7 health checking is enabled for the port, the port will fail the health check.

See "Changing the Health-Check Type" below.

---

For some ports, you also can customize the Layer 7 information sent with the health check. Here is the syntax.

**Syntax:** [no] protocol http | 80  
[url "[GET | HEAD] [/]<URL-page-name>" |  
port http status\_code <range> [<range> [<range> [<range>]]] |  
content-match <matching-list-name>]

This command changes one of the following HTTP health-check parameters. To change more than one of these parameters, enter a separate **protocol http** or **protocol 80** command for each parameter.

- **url** "[GET | HEAD] [/]<URL-page-name>" – This parameter specifies whether the HTTP health check performs a GET request or a HEAD request. For GET requests, you can specify the page that is requested. By default, a GET request asks for page "1.0".
- **port http status\_code** <range> [<range> [<range> [<range>]]] – This parameter changes the HTTP status codes that the ServerIron will accept as valid responses. Each <range> specifies the low number and high number in a range of status codes. You can specify up to four ranges (total of eight values). To specify a single message code for a range, enter the code twice. For example to specify 200 only, enter the following command: **port http status\_code 200 200**. For SLB, the default status code range is 200 – 299. If the server's reply to the health check contains a status code within this range, the ServerIron considers the HTTP application to be healthy.
- **content-match** <matching-list-name> – This parameter attaches a match list for an HTTP content verification

health check to the real server. An HTTP content verification health check is a type of Layer 7 health check in which the ServerIron examines text in an HTML file sent by a real server in response to an HTTP keepalive request. The ServerIron searches the text in the HTML file for user-specified selection criteria and determines whether the HTTP port on the real server is alive based on what it finds. The selection criteria used in HTTP content verification is contained in a matching list that is attached to one or more real servers. The following is an example of the commands used to set up a matching list. For information on how to configure the match lists, see "Configuring HTTP Content Matching Lists" on page 12-33.

**Syntax:** [no] protocol dns | 53 [addr\_query "<name>" | zone <zone-name>]

This command changes one of the following DNS health-check parameters. To change more than one of these parameters, enter a separate **protocol dns** or **protocol 53** command for each parameter.

- **addr\_query** "<name>" – This parameter specifies a domain name to be requested from the real server by the ServerIron. If the server successfully responds with the IP address for the domain name, the server passes the health check. There is no default.
- **zone** <zone-name> – This parameter specifies a DNS zone name. The ServerIron sends a Source-of-Authority (SOA) request for the zone name. If the server is authoritative for the zone and successfully responds to the SOA request, the server passes the health check. There is no default.

---

**NOTE:** If you do not configure one of these parameters, the DNS port will fail the health check.

---

**Syntax:** [no] protocol radius | 1812 [username <string>] | [password <string>] | [key <string>]

This command changes one of the following RADIUS health-check parameters. The health check requests values that are configured on the RADIUS server. To change more than one of these parameters, enter a separate **protocol radius** or **protocol 1812** command for each parameter.

- **username** <string> – This parameter specifies an authentication username on the server.
- **password** <string> – This parameter specifies an authentication password on the server.
- **key** <string> – This parameter specifies an authentication key on the server.

**Syntax:** [no] protocol ldap | 389 [<num>]

This command changes the LDAP version. The health check sent by the ServerIron differs depending on the version. You can specify 2 or 3. The default is 3.

### Using SSL Health Checks in a Health Check Policy

When SSL health checks are used in a health check policy, by default the simple SSL health check is used: The ServerIron sends the server an SSL client hello with the SSL SID set to 0; if the server responds, it passes the health check. However, if you use the **protocol ssl use-complete** command in a health check policy, it causes the ServerIron to negotiate an SSL connection and send a GET or HEAD request to the server.

For example, the following commands create a health check policy to test IP address 10.10.10.50, using SSL health checks.

```
ServerIron(config)# healthck check4 tcp
ServerIron(config-hc-check4)# dest-ip 10.10.10.50
ServerIron(config-hc-check4)# port ssl
ServerIron(config-hc-check4)# protocol ssl use-complete
ServerIron(config-hc-check4)# protocol ssl url "GET /secure.htm"
ServerIron(config-hc-check4)# protocol ssl status-code 200 200
ServerIron(config-hc-check4)# protocol ssl content-match m1
ServerIron(config-hc-check4)# l7-check
ServerIron(config-hc-check4)# enable
ServerIron(config-hc-check4)# exit
```

**Syntax:** [no] protocol ssl use-complete

### Changing the Health-Check Interval and Retries

By default, the ServerIron performs a health check every 5 seconds. If a reply is not received, the ServerIron will attempt the health check two more times before concluding that the application has failed the health check. You can change the number of seconds the ServerIron will wait for a reply to a health check and the number of retries.

---

**NOTE:** The number of retries is the total number of attempts the ServerIron will make. Thus, if you use the default interval and retries values, the ServerIron will send up to three health-check packets, at 5-second intervals. If a server does not respond within 15 seconds of the time the ServerIron sent the first health-check packet, the server fails the health check and the ServerIron concludes that the server is not available.

---

To change the interval for a health check, enter a command such as the following at the configuration level for the element-action expression that contains the health check:

```
ServerIron(config-hc-check1)# interval 30
```

**Syntax:** [no] interval <secs>

You can specify from 2 – 120 seconds. The default is 5 seconds.

To change the number of retries for a health check, enter a command such as the following at the configuration level for the element-action expression that contains the health check:

```
ServerIron(config-hc-check1)# retries 4
```

**Syntax:** [no] retries <num>

You can specify from 1 – 5 retries. The default is 3 retries.

---

**NOTE:** You also can globally change the interval and retries for an application port by editing its port profile. See “Adding a TCP or UDP Port, Specifying the Port Type, and Configuring the Keepalive Health Check” on page 12-23.

---

### Changing the Health-Check Type

For TCP application ports, you can change the health-check type between Layer 4 and Layer 7. By default, the ServerIron performs a Layer 7 health check in the following cases:

- The port is one of the following ports well-known to the ServerIron:
  - FTP – port 21. (Ports 20 and 21 both are FTP ports but on the ServerIron, the name “FTP” corresponds to port 21.)
  - HTTP – port 80
  - IMAP4 – port 143
  - LDAP – port 389
  - MMS – port 1755
  - NNTP – port 119
  - PNM – port 7070
  - POP3 – port 110
  - RTSP – port 554
  - SMTP – port 25
  - SSL – port 443
  - TELNET – port 23
- The port is not well-known to the ServerIron but you used the **protocol** command to specify the protocol of one of the well-known ports. By specifying the protocol, you configure the ServerIron to use the protocol's Layer 7 health-check method for the port.

If the TCP port is not one of the ports above or you did not specify a Layer 7 health-check method (using the **protocol** command), the ServerIron uses the Layer 4 health check for TCP.

---

**NOTE:** Changing the health-check type for UDP application ports has no effect. If the application port is RADIUS (1812) or DNS (53) or uses the health-check method of one of these ports, the ServerIron uses a Layer 7 health check. Otherwise, the ServerIron uses the Layer 4 health check for UDP.

---

The Layer 7 health-check methods differ depending on the application, and are described in “Health Check Summary” on page 12-4.

- TCP – The ServerIron attempts to engage in a normal three-way TCP handshake with the port on the real server:
  - The ServerIron sends a TCP SYN packet to the port on the real server.
  - The ServerIron expects the real server to respond with a SYN ACK.
  - If the ServerIron receives the SYN ACK, the ServerIron sends a TCP RESET, satisfied that the TCP port is alive.
- UDP – The ServerIron sends a UDP packet with garbage (meaningless) data to the UDP port.
  - If the server responds with an ICMP “Port Unreachable” message, the ServerIron concludes that the port is not alive.
  - If the server does not respond at all, the ServerIron assumes that the port is alive and received the garbage data. Since UDP is a connectionless protocol, the ServerIron and other clients do not expect replies to data sent to a UDP port. Thus, lack of a response is a good outcome.

```
ServerIron(config-hc-check1)# l4-check
```

The command in this example configures the ServerIron to use the Layer 4 health check for the application port in the element-action expression. Since the application port in this element-action expression is HTTP, the ServerIron will use the Layer 4 health check for TCP.

**Syntax:** [no] l4-check | l7-check

### Changing the Health-Check State

Once you configure an element-action expression, the health check in the expression is enabled by default. To disable the health check, enter the following command at the configuration level for the element-action expression:

```
ServerIron(config-hc-check1)# disable
```

**Syntax:** [no] disable | enable

---

**NOTE:** Health checking (keepalive) also must be enabled on the port profile level or the real server level. Otherwise, the health-check policy is used during initial bringup of the server but is not used for periodic health checks after the server is brought up.

---



---

**NOTE:** If the health check for an application on a server is disabled, the ServerIron assumes that the server and application are healthy and continues to send client requests to the server.

---



---

**NOTE:** If you change the health-check state from within the element-action expression, this state overrides the health-check state configured in the port profile for the application port or in the real server configuration.

---



---

**NOTE:** You can globally enable or disable all health-check policies. See “Globally Disabling All Health-Check Policies” on page 12-50.

---

### Configuring a Health-Check Policy

A health-check policy consists of one or more element-action expressions. When a logical expression contains multiple element-action expressions, the policy also contains the logical operator AND or OR.

You can use a health-check policy as an element-action expression in another policy.

To configure a health-check policy, enter commands such as the following:

```
ServerIron(config)# healthck "httpsrvr" boolean
ServerIron(config-hc-httpsrvr)# and "check1" "check2"
```

These commands configure a health-check policy that uses the element-action expressions "check1" and "check2". Since the AND operator is used, the real servers in both "check1" and "check2" must reply successfully for the health check to be successful. If only one of the servers replies, the health check is unsuccessful and the ServerIron stops using all the server application ports in the health-check policy "httpsrvr".

**Syntax:** [no] healthck "<policy-name>" boolean

**Syntax:** and | or "<element-name>" "<element-name>"

The <policy-name> parameter specifies the name of the health-check policy. The name can be up to 20 characters long. The name cannot contain blanks.

The **and** | **or** parameter specifies a logical operator in the health-check policy. You can enter two element-action expressions along with the logical operator **and** or **or**.

- If you specify **and**, the policy evaluates to true only if all elements (IP addresses) respond to the health check.
- If you specify **or**, the policy is true if at least one of the elements responds to the health check.

### **Configuring a Nested Health-Check Policy**

If you want to use a single health-check policy to test more than two IP addresses, configure health-check policies for all the IP addresses, and use them in another health-check policy. For example, to create a health-check policy that tests four IP addresses, enter commands such as the following:

```
ServerIron(config)# healthck check1 tcp
ServerIron(config-hc-check1)# dest-ip 10.10.10.50
ServerIron(config-hc-check1)# port http
ServerIron(config-hc-check1)# healthck check2 tcp
ServerIron(config-hc-check2)# dest-ip 10.10.10.20
ServerIron(config-hc-check2)# port http
ServerIron(config-hc-check2)# healthck check3 tcp
ServerIron(config-hc-check3)# dest-ip 10.10.10.30
ServerIron(config-hc-check3)# port http
ServerIron(config-hc-check3)# healthck check4 tcp
ServerIron(config-hc-check4)# dest-ip 10.10.10.40
ServerIron(config-hc-check4)# port http
```

The commands above configure four element-action expressions, one for each of four servers. The following commands configure two health-check policies, each of which contains two of the element-action expressions.

```
ServerIron(config-hc-check4)# healthck nested1 boolean
ServerIron(config-hc-nested1)# or check1 check2
ServerIron(config-hc-nested1)# healthck nested2 boolean
ServerIron(config-hc-nested2)# or check3 check4
```

The following command creates a health-check policy that contains the two policies configured above. The result is a single health-check policy for all four IP servers.

```
ServerIron(config-hc-nested2)# healthck checkall boolean
ServerIron(config-hc-checkall)# or nested1 nested2
```

In this example, the OR logical operator is used in all the policies. Thus, the "checkall" health check is successful if at least one of the four servers responds. To create more restrictive policies, you can use the AND logical operator. For example, if the AND operator is used in this configuration instead of OR, the health check is successful only if all four servers respond.

You also can combine policies that use AND with policies that use OR in nested health-check policies.

### **Attaching a Health-Check Policy to an Application Port on a Server**

After you configure logical expressions, you can attach them to application ports on real servers. **The ServerIron does not begin sending health-check packets until you attach the policy to a real server port.**

To attach a health-check policy to an application port on a server, enter commands such as the following:

```
ServerIron(config)# server real-name R1 10.10.10.50
ServerIron(config-rs-R1)# port 80 healthck "check1"
```

This command configures the ServerIron to base the health of application port 80 on real server R1 on the results of the check1 health-check policy.

### **Globally Disabling All Health-Check Policies**

You can easily disable all the health-check policies configured on the ServerIron. To do so, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# no server l4-check
```

---

**NOTE:** This command also disables the TCP and UDP Layer 4 health checks for all applications that are not associated with a health-check policy.

---

**Syntax:** [no] server l4-check

To re-enable the health-check policies, enter the following command.

```
ServerIron(config)# server l4-check
```

---

**NOTE:** The **server l4-check** command does not enable a policy if its element-action expressions contain the **disable** command. In this case, the policy remains disabled.

---

### **Displaying Health-Check Policy Information**

To display configuration information, current status, or statistics for health-check policies, use the CLI methods in the following sections.

#### **Displaying Health-Check Policies and Their Status**

To display a list of the configured health-check policies and their current status, enter the following command at any level of the CLI:

```
ServerIron(config-hc-check1)# show healthck
Total nodes: 6; Max nodes: 128
```

Name	Value	Enable	Type	Dest-IP	Port	Proto	Layer
check1	TRUE	YES	tcp	10.10.10.50	http	http	14-chk
check2	TRUE	YES	tcp	10.10.10.40	http	http	17-chk
check3	TRUE	NO	udp	10.10.10.30	http	http	14-chk
check4	TRUE	NO	udp	10.10.10.40	http	http	14-chk
check5	N/A	NO	udp	-	dns	dns	14-chk
httpsrvr	TRUE	YES	and	check1 check2			
nested1	N/A	na	and	check1 check2			
nested2	N/A	na	or	check3 check4			

**Syntax:** show healthck

This command shows the following information.

**Table 12.6: Health-Check Policy Status**

This Field...	Displays...
Total nodes	The number of health-check policies in the configuration. The number includes attached and unattached policies.
Max nodes	The maximum number of health-check policies you can configure.
Name	The element-action expression or policy name.
Value	<p>The current value of the policy. The value can be one of the following:</p> <ul style="list-style-type: none"> <li>TRUE – The most recent health check performed using this policy was successful. The ServerIron received a valid reply to the health check.</li> <li>FALSE – The most recent health check performed using this policy was unsuccessful.</li> <li>N/A (Not Attached) – The policy is not attached to a real server.</li> </ul> <p><b>Note:</b> If the policy is disabled, the value is always TRUE. This is because the ServerIron assumes a server is healthy unless its health check is enabled <b>and</b> the server has not responded appropriately to the health check.</p>
Enable	<p>The state of the policy, which can be one of the following:</p> <ul style="list-style-type: none"> <li>YES – The policy is enabled.</li> <li>NO – The policy is disabled.</li> <li>na (not applicable) – This field does not apply to the policy. This value indicates that the policy is not attached to a real server.</li> </ul>
Type	<p>The element-action expression or policy type, which can be one of the following:</p> <ul style="list-style-type: none"> <li>tcp – An element-action expression for a TCP application port.</li> <li>udp – An element-action expression for a UDP application port.</li> <li>and – A policy containing element-action expressions joined by AND.</li> <li>or – A policy containing element-action expressions joined by OR.</li> </ul>
Dest-IP	<p>For element-action expressions, the IP address of the real server. For policies, this field shows the element-action expressions in the policy.</p> <p>The value " - " indicates that the IP address has not been specified.</p>
Port	For element-action expressions, the application port. This field does not apply to policies.
Proto	<p>For element-action expressions, the health-check method to be used for the port.</p> <p><b>Note:</b> If the value is " - ", the protocol has not been specified and the port is not well-known to the ServerIron.</p>

**Table 12.6: Health-Check Policy Status (Continued)**

This Field...	Displays...
Layer	<p>The type of health check, which can be one of the following:</p> <ul style="list-style-type: none"> <li>• I4-chk – Layer 4 TCP or UDP health check.</li> <li>• I7-chk – Layer 7 application-specific health check.</li> </ul>

## Configuring Boolean Health-Check Policies (ServerIronXL)

You can configure a group of Layer 3 health checks as a health-check policy and associate the group with a specific TCP or UDP application port on an individual virtual IP address (VIP). You can use health-check policies to check the health of downstream routers and bring a VIP down if a router fails a health check.

**NOTE:** This section applies only to software release 07.3.02 and higher for the ServerIronXL. To use Boolean health-check policies on the ServerIron 400 or ServerIron 800, see “Configuring Boolean Health-Check Policies (ServerIron 400 and ServerIron 800)” on page 12-41.

Depending on the conditions you specify when you configure a health-check policy, the ServerIron will bring the VIP down in one of the following cases:

- Any one of the health checks fails (AND condition)
- All of the health checks fail (OR condition)

For finer control, you can combine OR and AND conditions.

The health checks consist of an ICMP echo packet, the same type of packet used by the ServerIron for other Layer 3 health checks. By default, the ServerIron performs the VIP-based health checks you have configured by sending an ICMP ping to the specified IP address every 400 milliseconds.

- If the ServerIron receives one or more responses within 1.2 seconds, the ServerIron concludes that the VIP is healthy.
- Otherwise, the ServerIron reattempts the health check by sending another ping. The ServerIron reattempts an unanswered health check up to two more times before concluding that the VIP is unhealthy.

**NOTE:** If a VIP fails a health check, the ServerIron brings the VIP down. However, the VIP’s state listed by the **show server virtual** command remains “enabled”.

## Configuring a Health-Check Policy

Health-check policies consist of element-action expressions and logical operators.

- Element-action expression – In the case of Layer 3 health checks, an element-action expression consists of the IP protocol to be used (ICMP) and the IP address to be checked.
- Logical operator – A logical operator is the Boolean operator OR or AND. To configure a health-check policy that requires a reply from all IP addresses in the policy, use the operator AND. To create a policy that is successful if at least one of the addresses replies, use OR.

You can use the same element-action expressions in multiple logical expressions if desired. You can configure up to 254 health-check policies. The default maximum number you can configure is 128. You can change the maximum to a number from 64 – 254.

To use a health-check policy:

- Configure the element-action expressions.
- Configure the health-check policy using element-action expressions and the logical operator AND or OR.



- Bind logical expressions to application ports on specific VIPs. A health check policy does not take effect until you bind it to an application port on a VIP.

Here is an example of how to configure and apply a Layer 3 health-check policy. The example is described in detail in the following sections.

```
ServerIron(config)# healthck Rtr2-ck1 icmp
ServerIron(config-hc-Rtr2-ck1)# dest-ip 10.168.2.56
ServerIron(config-hc-Rtr2-ck1)# healthck Rtr2-ck2 icmp
ServerIron(config-hc-Rtr2-ck2)# dest-ip 10.168.2.57
ServerIron(config)# healthck Router2 boolean
ServerIron(config-hc-Router2)# and Rtr2-ck1 Rtr2-ck2
ServerIron(config)# server virtual-name VIP1 1.1.1.1
ServerIron(config-vs-VIP1)# port http healthck Router2
```

These commands configure two element-action expressions, "Rtr2-ck1" and "Rtr2-ck2", and use them in a health-check policy called "Router2". The last two commands apply the health-check policy to the HTTP port on VIP1. For more information, see the following sections.

### **Configuring an Element-Action Expression**

For Layer 3 health-check policies, an element-action expression contains an IP address. To configure an element-action expression, enter commands such as the following:

```
ServerIron(config)# healthck Rtr2-ck1 icmp
ServerIron(config-hc-Rtr2-ck1)# dest-ip 10.168.2.56
ServerIron(config-hc-Rtr2-ck1)# healthck Rtr2-ck2 icmp
ServerIron(config-hc-Rtr2-ck2)# dest-ip 10.168.2.57
```

The commands in this example configure two element-action expressions.

**Syntax:** [no] healthck <element-name> <protocol>

**Syntax:** [no] dest-ip <ip-addr>

The <element-name> parameter specifies a name for the element-action expression. The name can be up to 20 characters long. The name cannot contain blanks.

The <protocol> parameter specifies the IP protocol to use for the health. The Layer health checks use ICMP echo packets. Therefore, you must specify **icmp**.

The <ip-addr> parameter specifies the IP address to check.

### **Configuring a Health-Check Policy**

A health-check policy consists of one or more element-action expressions. When a logical expression contains multiple element-action expressions, the policy also contains the logical operator AND or OR.

You can use a health-check policy as an element-action expression in another policy.

To configure a health-check policy, enter commands such as the following:

```
ServerIron(config)# healthck Router2 boolean
ServerIron(config-hc-Router2)# and Rtr2-ck1 Rtr2-ck2
```

These commands configure a health-check policy that uses the element-action expressions "Rtr2-ck1" and "Rtr2-ck2". Since the AND operator is used, the IP addresses in both "Rtr2-ck1" and "Rtr2-ck2" must reply successfully for the health check to be successful. If only one of the addresses replies, the health check is unsuccessful and the ServerIron brings the VIP down.

**Syntax:** [no] healthck <policy-name> boolean

**Syntax:** <element-name>

Or

**Syntax:** and | or <element-name> <element-name>

The <policy-name> parameter specifies the name of the health-check policy. The name can be up to 20 characters long. The name cannot contain blanks.

The **and** or **or** parameter specifies a logical operator in the health-check policy.

- You can specify an element-action without also specifying a logical operator (AND or OR). In this case, the policy checks the health of the specified element (IP address) and has a true result (the health check is successful) if the element replies to the health check.
- You can enter two element-action expressions along with the logical operator **and** or **or**.
  - If you specify **and**, the policy evaluates to true only if all elements (IP addresses) respond to the health check.
  - If you specify **or**, the policy is true if at least one of the elements responds to the health check.

### **Configuring a Nested Health-Check Policy**

If you want to use a single health-check policy to test more than two IP addresses, configure health-check policies for all the IP addresses, and use them in another health-check policy. For example, to create a health-check policy that tests four IP addresses, enter commands such as the following:

```
ServerIron(config)# healthck nest1 icmp
ServerIron(config-hc-nest1)# dest-ip 1.1.1.10
ServerIron(config-hc-nest1)# healthck nest2 icmp
ServerIron(config-hc-nest2)# dest-ip 1.1.1.20
ServerIron(config-hc-nest2)# healthck nest3 icmp
ServerIron(config-hc-nest3)# dest-ip 1.1.1.30
ServerIron(config-hc-nest3)# healthck nest4 icmp
ServerIron(config-hc-nest4)# dest-ip 1.1.1.40
```

The commands above configure four element-action expressions, one for each IP address. The following commands configure two health-check policies, each of which contains two of the IP addresses.

```
ServerIron(config-hc-nest4)# healthck nested1 boolean
ServerIron(config-hc-nested1)# or nest1 nest2
ServerIron(config-hc-nested1)# healthck nested2 boolean
ServerIron(config-hc-nested2)# or nest3 nest4
```

The following command creates a health-check policy that contains the two policies configured above. The result is a single health-check policy for all four IP addresses.

```
ServerIron(config-hc-nested2)# healthck check1 boolean
ServerIron(config-hc-check1)# or nested1 nested2
```

In this example, the OR logical operator is used in all the policies. Thus, the "check1" health check is successful if at least one of the four IP addresses responds. To create more restrictive policies, you can use the AND logical operator. For example, if the AND operator is used in this configuration instead of OR, the health check is successful only if all four IP addresses respond.

You also can combine policies that use AND with policies that use OR in nested health-check policies.

### **Binding a Health-Check Policy to an Application Port on a VIP**

After you configure logical expressions, you can bind them to application ports on VIPs. A health-check policy does not take effect until you bind the policy to an application port on a VIP.

To bind a health-check policy to an application port on a VIP, enter commands such as the following:

```
ServerIron(config)# server virtual-name VIP1 1.1.1.1
ServerIron(config-vs-VIP1)# port http healthck Router2
```

This command configures virtual IP address VIP1 to use the health-check policy named "Router2" to check the health of HTTP (port 80) for the VIP.

**Syntax:** [no] port <tcp/udp-portnum> healthck <policy-name>

The <tcp/udp-portnum> parameter specifies a TCP or UDP application port. Specify the port number or one of the following well-known port names:

- **dns** – port 53

- **ftp** – port 21. (Ports 20 and 21 both are FTP ports but in the ServerIron, the name “ftp” corresponds to port 21.)
- **http** – port 80
- **imap4** – port 143
- **ldap** – port 389
- **nntp** – port 119
- **ntp** – port 123
- **pop2** – port 109
- **pop3** – port 110
- **radius** – UDP port 1812
- **radius-old** – the ServerIron name for UDP port 1645, which is used in some older RADIUS implementations instead of port 1812
- **smtp** – port 25
- **snmp** – port 161
- **ssl** – port 443
- **telnet** – port 23
- **tftp** – port 69

The <policy-name> parameter specifies the health-check policy you want to use to check the Layer 3 health of a device associated with the application port.

### Changing the Memory Allocation for Health-Check Policies

By default, you can configure up to 128 health-check policies. To change the maximum number of health-check policies that can be configured on the ServerIron, enter commands such as the following:

```
ServerIron(config)# system-max healthck 254
ServerIron(config)# end
ServerIron# reload
```

**Syntax:** [no] system-max healthck <num>

The <num> parameter specifies the maximum number of health-check policies and can be a number from 64 – 254. The default is 128.

---

**NOTE:** You must reload the software to place the change into effect.

---

### Displaying Health-Check Policy Information

To display configuration information, current status, or statistics for health-check policies, use the CLI methods in the following sections.

#### Displaying Health-Check Policies and Their Status

To display a list of the configured health-check policies and their current status, enter the following command at any level of the CLI:

```
ServerIron(config)# show healthck
Total nodes: 4; Max nodes: 128
      Name      Value      Type
-----
Rtr1-ck1       N/B       icmp 10.168.2.46
Rtr1-ck2       N/B       icmp 10.168.2.47
Router1        N/B       or Rtr1-ck1 Rtr1-ck2
Rtr2-ck1       TRUE      icmp 10.168.2.56
```

```

Rtr2-ck2    TRUE    icmp 10.168.2.57
Router2     TRUE    and Rtr2-ck1 Rtr2-ck2
Rtr3-ck1    FALSE   icmp 10.168.2.66
Rtr3-ck2    TRUE    icmp 10.168.2.67
Router3     FALSE   and Rtr3-ck1 Rtr3-ck2

```

**Syntax:** show healthck

This command shows the following information.

**Table 12.7: Health-Check Policy Status**

This Field...	Displays...
Total nodes	The number of health-check policies in the configuration. The number includes bound and unbound policies.
Max nodes	The maximum number of health-check policies you can configure. <b>Note:</b> To change this amount, see “Changing the Memory Allocation for Health-Check Policies” on page 12-55.
Name	The policy name.
Value	The current value of the policy. The value can be one of the following: <ul style="list-style-type: none"> <li>TRUE – The most recent health check performed using this policy was successful. The ServerIron received a valid reply to the health check.</li> <li>FALSE – The most recent health check performed using this policy was unsuccessful.</li> <li>N/B – The health check is not bound to any VIP and thus is not in use.</li> </ul>
Type	The element-action expression in the policy. For Layer 3 health checks, this information consists of the protocol (ICMP) and the IP address tested by the health check.

#### **Displaying and Clearing Health-Check Policy Statistics**

To display health-check policy statistics, enter the following command at any level of the CLI:

```

ServerIron(config)# show healthck statistics
Ping Statistics:
Sent: 1524                      Received: 1524
Invalid Replies: 0              Dropped Replies: 0

```

**Syntax:** show healthck statistics

This command shows the following information.

**Table 12.8: Health-Check Policy Statistics**

This Field...	Displays...
Sent	The number of health-check packets sent by bound health-check policies.

**Table 12.8: Health-Check Policy Statistics (Continued)**

This Field...	Displays...
Received	The number of replies received. A received reply results in a true condition.  <b>Note:</b> Since the ServerIron retries a health check if a reply is not received, a higher sent count than receive count does not necessarily indicate a problem.
Invalid Replies	The number of replies that were received that had an invalid ID. The ServerIron is sometimes able to resolve an invalid ID. If the ServerIron cannot resolve the invalid ID, the device drops the reply and increments the Dropped Replies counter.
Dropped Replies	The number of replies that the ServerIron dropped.

To clear health-check policy statistics, enter the following command:

```
ServerIron(config)# clear healthck statistics
```

**Syntax:** clear healthck statistics

## Viewing Application Port Status in the Syslog

The ServerIron generates Syslog messages for changes to the Layer 4 or Layer 7 status of a real server. To display the Syslog buffer on the ServerIron, enter the following command:

```
ServerIron(config)# show log
Dynamic Log Buffer (50 entries):
03d02h47m38s:N:L4 server 192.168.1.170 danPC is down
03d02h46m18s:N:L4 server 192.168.1.170 danPC is up
03d02h46m08s:I:Interface ethernet5, state up
```

This example shows log entries for a real server named "danPC" with IP address 192.168.1.170. In this example, the real server passed a Layer 4 or Layer 7 health check ("up"), but then failed a Layer 4 or Layer 7 health check ("down") later.

**Syntax:** show logging

**NOTE:** The log messages do not distinguish between Layer 4 and Layer 7 health checks. When the status changes based on either type of health check, the ServerIron logs the event as shown in this example.

## Configuring Session Table Parameters

The ServerIron maintains state information for TCP and UDP connections in the session table. The **session table** contains an entry for each TCP and UDP session between the ServerIron and a client or real server. The ServerIron uses the session table entries for health checks, stateful failover in hot-standby configurations, and other functions.

Each entry in the session table is a **session**. A session consists of the following:

- Source IP address
- Source application port
- Destination IP address
- Destination application port
- Protocol (TCP or UDP)

A **connection** consists of two sessions, a send session and a receive session. For example, a TCP connection between a client and a server consists of two sessions, a client-to-server session and a server-to-client session.

---

**NOTE:** "Stateless" features such as stateless application ports and stateless health checks do not use session table entries.

---

This section describes how to configure the following session table parameters:

- Maximum number of sessions
- Maximum age of TCP session entries
- Maximum age of UDP session entries
- Clock scale for TCP and UDP session age timers
- Logging of session table entries

## Modifying Maximum Session Limit

You can limit the maximum number of total active sessions the ServerIron allows. An active session is a session entry in the ServerIron's session table. Thus, a UDP or TCP session that has become idle but has not yet timed out (according to the UDP or TCP age timer) is an "active" session in this table.

Up to 2,000,000 sessions are supported on a ServerIron 400 or ServerIron 800; up to 1,000,000 sessions are supported on a ServerIron with 32MB memory installed; and up to 160,000 sessions are supported on a ServerIron with 8MB of memory. Possible values for ServerIron 400 and ServerIron 800 systems are 32,768 – 2,000,000 with a default value of 1,000,000. Possible values for 32MB systems are 32,768 – 1,000,000 with a default value of 524,288. Possible values for 8MB systems are 32,768 – 160,000 with a default value of 131,072.

---

**NOTE:** To place this change into effect, you must save the change to the startup-config file, then reload the software.

---

### USING THE CLI

To modify the maximum sessions supported on all servers, enter commands such as the following:

```
ServerIron(config)# server session-limit 50000
ServerIron(config)# write memory
ServerIron(config)# end
ServerIron# reload
```

**Syntax:** server session-limit <value>

The <value> for ServerIron 400 and ServerIron 800 systems can be from 32768 – 2000000. The <value> for 32M systems can be from 32768 – 1000000. The default is 1000000.

On 8M systems, the <value> can be from 32768 – 160000. The default is 131072.

### USING THE WEB MANAGEMENT INTERFACE

To modify the maximum sessions supported on a switch:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu.
5. Enter a value from 32768 – 2000000 in the Max Session Limit field.
  - Possible values for ServerIron 400 and ServerIron 800 systems are 32768 – 2000000 with a default value of 1000000.

- Possible values for 32MB systems are 32768 – 1000000 with a default value of 1000000.
  - Possible values for 8MB systems are 32768 – 160000 with a default value of 131072.
6. Select the Apply button to assign the change.
  7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.
  8. Select the option to reload the software.

## Modifying the TCP Age

The TCP age specifies how many minutes a TCP server connection can remain inactive before the ServerIron times out the session. Possible values are from 2 – 60 minutes. The default is 30 minutes.

If you change the TCP age, the change affects only new TCP sessions that start after you make the change. The maximum age for sessions that are already in the session table does not change.

---

**NOTE:** This parameter globally sets the age for all TCP ports. To override the setting for an individual TCP port, change that port's profile. See "Overriding the Global TCP or UDP Age" on page 12-27.

---

### USING THE CLI

To modify the server TCP age to 20 minutes from the default value of 30 minutes, enter the following command:

```
ServerIron(config)# server tcp-age 20
```

**Syntax:** server tcp-age <2-60>

You can specify from 2 – 60 minutes.

### USING THE WEB MANAGEMENT INTERFACE

To modify the TCP aging out parameter:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu.
5. Enter a value from 2 – 60 in the TCP Age field.
6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Modifying the UDP Age

You can modify the aging out parameter for inactive UDP server connections. Possible values are from 2 – 60 minutes. The default is 5 minutes.

---

**NOTE:** This parameter globally sets the age for all UDP ports. To override the setting for an individual TCP port, change that port's profile. See "Overriding the Global TCP or UDP Age" on page 12-27.

---

---

**NOTE:** The ServerIron immediately deletes a UDP DNS or RADIUS session table entry when the ServerIron receives a reply for the application from a real server. If desired, you can configure the ServerIron to age these ports like other UDP ports, using the UDP age timer. See "Normal UDP Aging for DNS and RADIUS" on page 6-60.

---

### USING THE CLI

To modify the server UDP age to 20 minutes from the default value of 5 minutes, enter the following command:

```
ServerIron(config)# server udp-age 20
```

**Syntax:** server udp-age <2-60>

You can specify from 2 – 60 minutes.

#### [USING THE WEB MANAGEMENT INTERFACE](#)

To modify the UDP aging out parameter:

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu.
5. Enter a value from 2 – 60 in the UDP Age field.
6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Modifying the Clock Scale

The ServerIron uses a configurable clock scale for the following session timers:

- TCP age
- UDP age

You can adjust the clock scale for configurations that require TCP or UDP timeouts longer than the maximum configurable value (60 minutes). For example, if you set the clock scale to 2, the TCP and UDP age timer values are multiplied by 2. Thus, a TCP age of 60 would then be equivalent to 120 minutes instead of 60 minutes.

You can set the clock scale to a value from 1 – 20. The default is 1.

#### [USING THE CLI](#)

To change the clock scale, enter a command such as the following:

```
ServerIron(config)# server clock-scale 2
```

**Syntax:** server clock-scale <multiplier>

The <multiplier> can be a value from 1 – 20. The default is 1.

#### [USING THE WEB MANAGEMENT INTERFACE](#)

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to SLB in the tree view to expand the list of Server Load Balancing option links.
4. Select the General link from the menu.
5. Enter a value from 0 – 20 in the Clock Scale field. The default is 1.
6. Select the Apply button to assign the change.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Enabling Syslog Messages for Session Table Entries

You can configure the ServerIron to send a message to external Syslog servers when the software creates a session table entry. The messages indicate the following information:

- Source IP address



- Source TCP or UDP application port
- Destination IP address
- Destination TCP or UDP application port
- Layer 4 protocol (TCP or UDP)
- Message time (measured in units of 100 milliseconds, relative to system uptime)
- URL (optional)
- Cookie (optional)
- Internet (applies to TCS only)

You can enable TCP/UDP logging on a global basis for all TCP and UDP ports or for individual TCP or UDP ports.

When you enable TCP/UDP logging, you can specify whether all new session table entries generate log messages or only the entries that are used for Source NAT.

In addition, you can enable logging for URL or Cookie information. The URL logging option applies only when URL switching is enabled. The Cookie logging option applies only when Cookie switching is enabled.

Here is an example of a Syslog message for a session:

```
src-ip = 192.168.002.032   src-port = 00197   dst-ip = 192.168.002.012
dst-port = 00080   protocol = TCP   time =0000078656   Url = abcdefghijklmnop
Cookie = qrstuvwxyz   Internet
```

The "Internet" parameter at the end of the message applies only to TCS, and indicates that the ServerIron sent the client request to the Internet instead of to a cache server.

The time value in this example is in the format for devices on which the system time add date have not been set. For information, see the *Foundry ServerIron Command Line Interface Reference*.

---

**NOTE:** The feature description and command syntax use the terms "session" and "connection". A connection consists of multiple sessions, for the send and receive directions.

---

---

**NOTE:** Since the log messages are generated when the software creates a session table entry, features that do not use session table entries do not result in log messages. For example, if you configure a TCP or UDP port to be stateless, the ServerIron does not create session table entries for the port and therefore does not generate log messages for the port.

---

## Enabling TCP/UDP Session Logging

You can enable session logging globally for all ports or on an individual TCP or UDP port basis.

### *Globally Enabling TCP/UDP Session Logging*

To enable session logging for all TCP and UDP ports, use the following CLI method.

#### *USING THE CLI*

To enable session logging for all TCP and UDP ports, enter a command such as the following:

```
ServerIron(config)# server connection-log all
```

The command in this example enables logging for all new session table entries. To enable logging only for new sessions that are used for Source NAT, enter the following command:

```
ServerIron(config)# server connection-log src-nat
```

**Syntax:** server connection-log all | src-nat [url] [cookie]

The **all** parameter enables logging for all sessions.

The **src-nat** parameter enables logging only for sessions that are used for Source NAT.

The **url** parameter enables logging of URL information for sessions that contain a URL.

The **cookie** parameter enables logging of Cookie information for sessions that contain a Cookie.

---

**NOTE:** The URL logging option applies only when URL switching is enabled. The Cookie logging option applies only when Cookie switching is enabled.

---

#### USING THE WEB MANAGEMENT INTERFACE

You cannot configure this feature using the Web management interface.

#### Enabling Session Logging for an Individual TCP or UDP Port

To enable session logging for a specific TCP or UDP port, use the following CLI method.

#### USING THE CLI

To enable session logging for a specific TCP or UDP port, enter commands such as the following:

```
ServerIron(config)# server port 80
ServerIron(config-port-80)# connection-log all url cookie
```

**Syntax:** connection-log all | src-nat [url] [cookie]

The parameter values are the same as the values for globally enabling logging.

#### USING THE WEB MANAGEMENT INTERFACE

You cannot configure this feature using the Web management interface.

## Configuring the Slow-Start Mechanism

When the ServerIron begins sending client requests to a real server that has recently gone online, it allows the server to ramp up by using the **slow-start mechanism**. The slow-start mechanism allows a server (or a port on the server) to handle a limited number of connections at first and then gradually handle an increasing number of connections until the maximum is reached.

The ServerIron uses two kinds of slow-start mechanisms:

- The non-configurable **server slow-start mechanism** applies to a real server that has just gone online
- The configurable **port slow-start mechanism** applies to individual TCP application ports that have just been activated on a real server

### Overview

The ServerIron uses the server slow-start mechanism to adjust the maximum number of connections that can be established for a real server that has just gone online. The ServerIron begins with a connection limit that is lower than the maximum configured value (which is one million by default) and gradually increases this connection limit until the maximum configured value is reached.

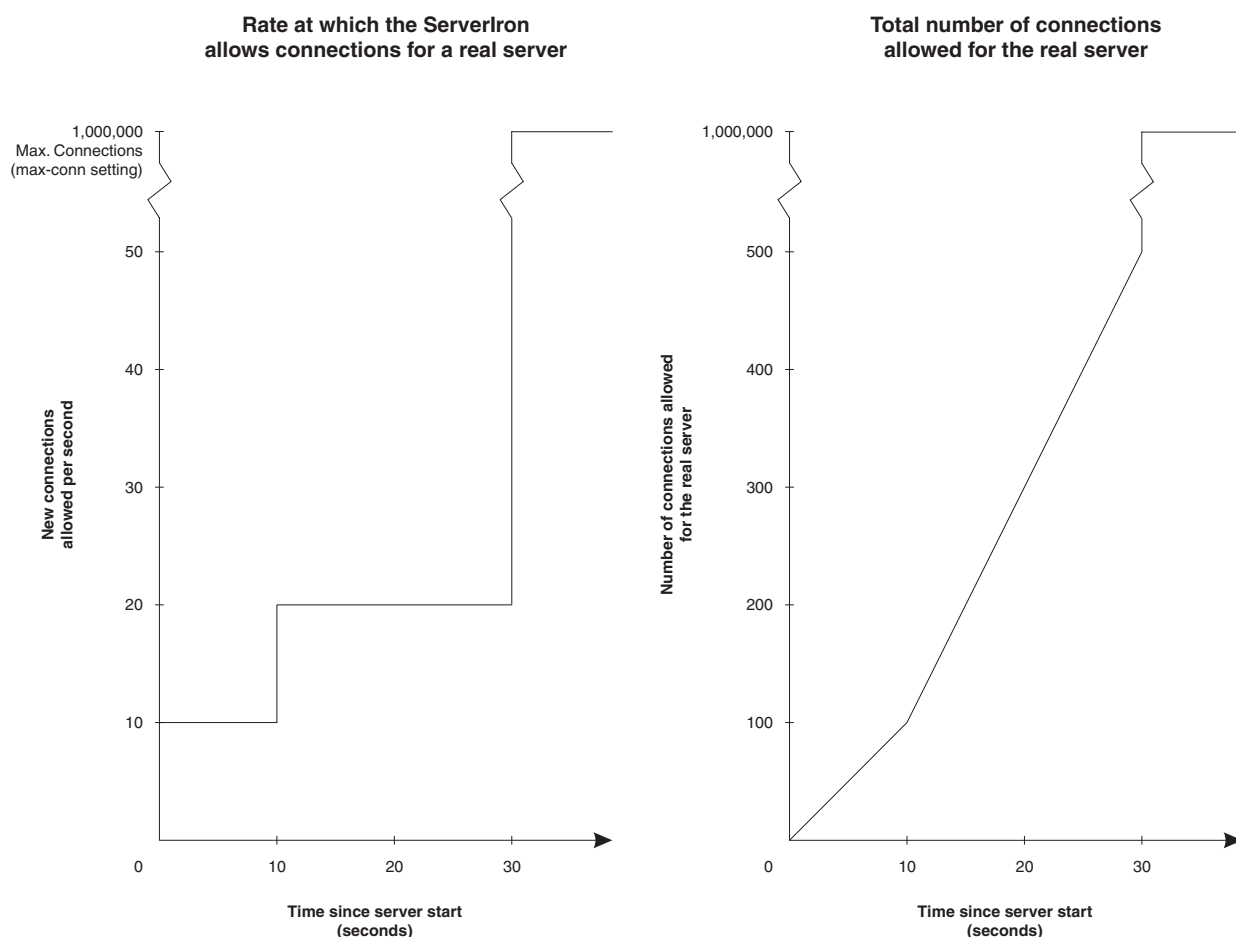
The server slow-start mechanism is especially useful when least connections is the distribution predictor. Without the server slow-start mechanism, a server that is just brought online could receive all the new connections in a flurry, which could bring the server down. Many servers cannot handle more than 2,000 new connections per second.

---

**NOTE:** The server slow-start mechanism is always applied to all real servers when they are brought online. Unlike the slow-start mechanism for individual ports, described in the next section, the server slow-start mechanism is not configurable.

---

The two graphs in Figure 12.3 illustrate how the server slow-start mechanism ramps up the connections for a real server during the 30-second slow-start period. The graph on the left shows the rate at which the number of connections increases over the slow-start period. The graph on the right shows how the maximum number of connections the ServerIron allows for the real server increases over the slow-start period.

**Figure 12.3** Slow-start mechanism for a real server

The graph on the left shows the rate at which the ServerIron allows connections for a given real server, as follows:

- From the time the real server is brought online until 10 seconds afterwards, the ServerIron allows the real server up to 10 new connections every second.
- From 10 seconds to 30 seconds, the ServerIron allows up to 20 new connections every second.
- After 30 seconds, the connection flow control delivered by the slow-start mechanism ends, and the ServerIron allows up to the maximum number of connections to the server. The maximum number of allowed connections for a real server is set by the **max-conn** command; this is one million connections by default.

The graph on the right shows how the maximum number of connections allowed for the real server increases over the 30-second slow-start period. The following table lists the maximum number of connections a real server can have during each second of the slow-start period.

Seconds after going online	Max. Connections	Seconds after going online	Max. Connections
1	10	16	220
2	20	17	240
3	30	18	260

Seconds after going online	Max. Connections	Seconds after going online	Max. Connections
4	40	19	280
5	50	20	300
6	60	21	320
7	70	22	340
8	80	23	360
9	90	24	380
10	100	25	400
11	120	26	420
12	140	27	440
13	160	28	460
14	180	29	480
15	200	30	500

When the slow-start period ends after 30 seconds, the maximum number of connections a real server can have is determined by the **max-conn** setting for the real server; this is one million connections by default.

## Port Slow-Start Mechanism

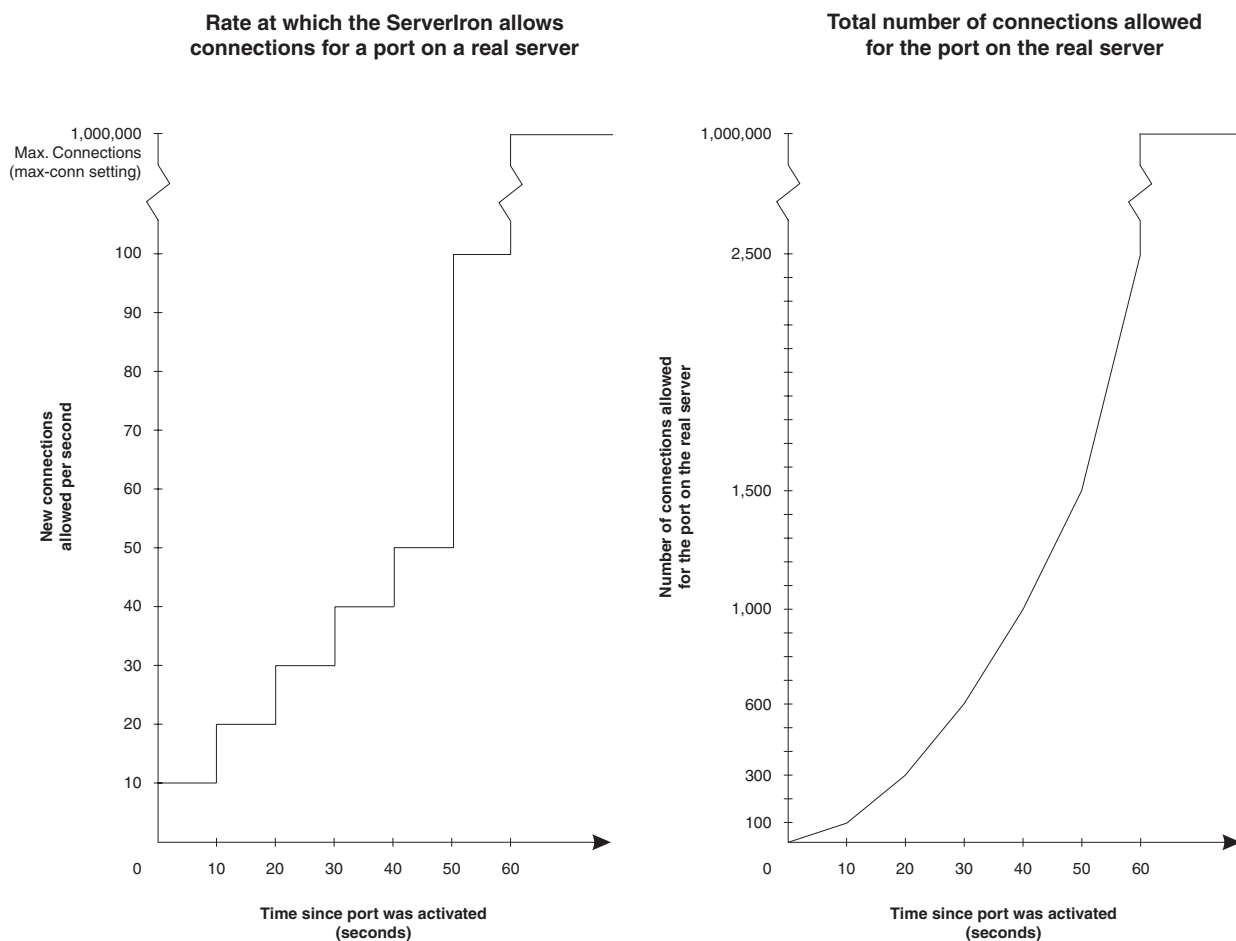
When individual TCP application ports on a real server are activated, they are allocated connections using the port slow-start mechanism, which works differently from the server slow-start mechanism described in the previous section.

When a port on a real server becomes active, the ServerIron applies the **default slow-start mechanism** to regulate how fast connections for the port are established. In addition, you can set up a **user-configured slow-start mechanism** that regulates how fast connections are established for specific ports on specific real servers. The following sections explain how the default slow-start mechanism works, as well as how to set up a user-configured slow-start mechanism and apply it to a port on a real server.

### Default Port Slow-Start Mechanism

By default, when a port is activated, the ServerIron gives it 60 seconds of warm-up time. Over this period, the ServerIron gradually increases the number of connections it allows for the port. The default slow-start mechanism is always applied to all ports when they are first brought online, unless they are configured to use a user-configured slow-start mechanism.

The two graphs in Figure 12.4 illustrate how the default slow-start mechanism ramps up the connections for a port on a real server. The graph on the left shows the rate at which the number of connections increases over the slow-start period. The graph on the right shows how the maximum number of connections the ServerIron allows for the port on the real server increases over the slow-start period.

**Figure 12.4** Default slow-start mechanism for a port

The graph on the left shows the rate at which the ServerIron allows connections for a given port on a real server, as follows:

- From the time the port is activated until 10 seconds afterwards, the ServerIron allows the port up to 10 new connections every second.
- From 10 seconds to 20 seconds, the ServerIron allows up to 20 new connections every second.
- From 20 seconds to 30 seconds, the ServerIron allows up to 30 new connections every second.
- From 30 seconds to 40 seconds, the ServerIron allows up to 40 new connections every second.
- From 40 seconds to 50 seconds, the ServerIron allows up to 50 new connections every second.
- From 50 seconds to 60 seconds, the ServerIron allows up to 100 new connections every second.
- After 60 seconds, the connection flow control delivered by the slow-start mechanism ends, and the ServerIron allows up to the maximum number of connections for the port on the server. The maximum number of allowed connections for a real server is set by the **max-conn** command; this is one million connections by default.

The graph on the right shows how the maximum number of connections allowed for the port on the real server increases over the slow-start period. The following table lists the maximum number of connections a port can have at 10-second intervals.

Seconds after port activated	Max. Connections
10	100
20	300
30	600
40	1,000
50	1,500
60	2,500

When the slow-start period ends after 60 seconds, the maximum number of connections a port on a real server can have is determined by the **max-conn** setting for the real server; this is one million connections by default.

### User-Configured Port Slow-Start Mechanism

You can configure how fast the ServerIron ramps up a particular port on a particular real server by setting up a user-configured slow-start mechanism. Unlike the default port slow-start mechanism, which applies to all ports on all real servers, a user-configured slow-start mechanism is applied to a specific port on a specific real server.

A user-configured slow-start mechanism sets the rate at which the ServerIron allows connections for a port over two configurable intervals (which comprise the slow-start period), as well as a limit for the total number of connections that the port on the real server can have during the time the server is active.

Setting up a user-configured slow-start mechanism consists of two steps:

1. Setting up a **slow-start list** for a port
2. Applying the slow-start list to a port on a real server

For example, the following commands set up a slow-start list for port 80 (HTTP):

```
ServerIron(config)# server port 80
ServerIron(config-port-80)# slow-start 101 10 30 20 30 600
ServerIron(config-port-80)# exit
```

**Syntax:** slow-start <list-id> <rate1> <interval1> <rate2> <interval2> <max-connections>

In the **slow-start** command, the <list-id> parameter identifies the slow-start list. This ID can be a number from 1 – 1000000. When you apply the slow-start list to a port on a real server, you refer to the slow-start list by this ID number. You can create multiple slow-start lists for a given port and assign them each an ID number.

The <rate1> parameter specifies the number of connections per second allowed for the port during the first interval. This can be a number from 1 – 1000000. From the time the port is activated until the end of the first interval, the ServerIron allows the port on the real server up to this number of new connections every second.

The <interval1> parameter specifies the length of the first interval in seconds. This can be a number from 1 – 1000000.

The <rate2> parameter specifies the number of connections per second allowed for the port during the second interval. This can be a number from 1 – 1000000. From the end of the first interval until the end of the second interval, the ServerIron allows the port on the real server up to this number of new connections every second.

The <interval2> parameter specifies the length of the second interval in seconds. This can be a number from 1 – 1000000. The number of seconds in the first interval, plus the number of seconds in the second interval, comprise

the slow-start period. In this example, <interval1> is 30 seconds, and <interval2> is 30 seconds, so the slow-start period is 60 seconds.

The <max-connections> parameter sets a ceiling for the number of concurrent connections allowed for the port during the time the server is active. This can be a number from 1 – 1000000. No more than this number of connections can be established for the port on the real server where this slow-start mechanism is applied.

Once you have created a slow-start list, you apply it to a port on a real server; for example:

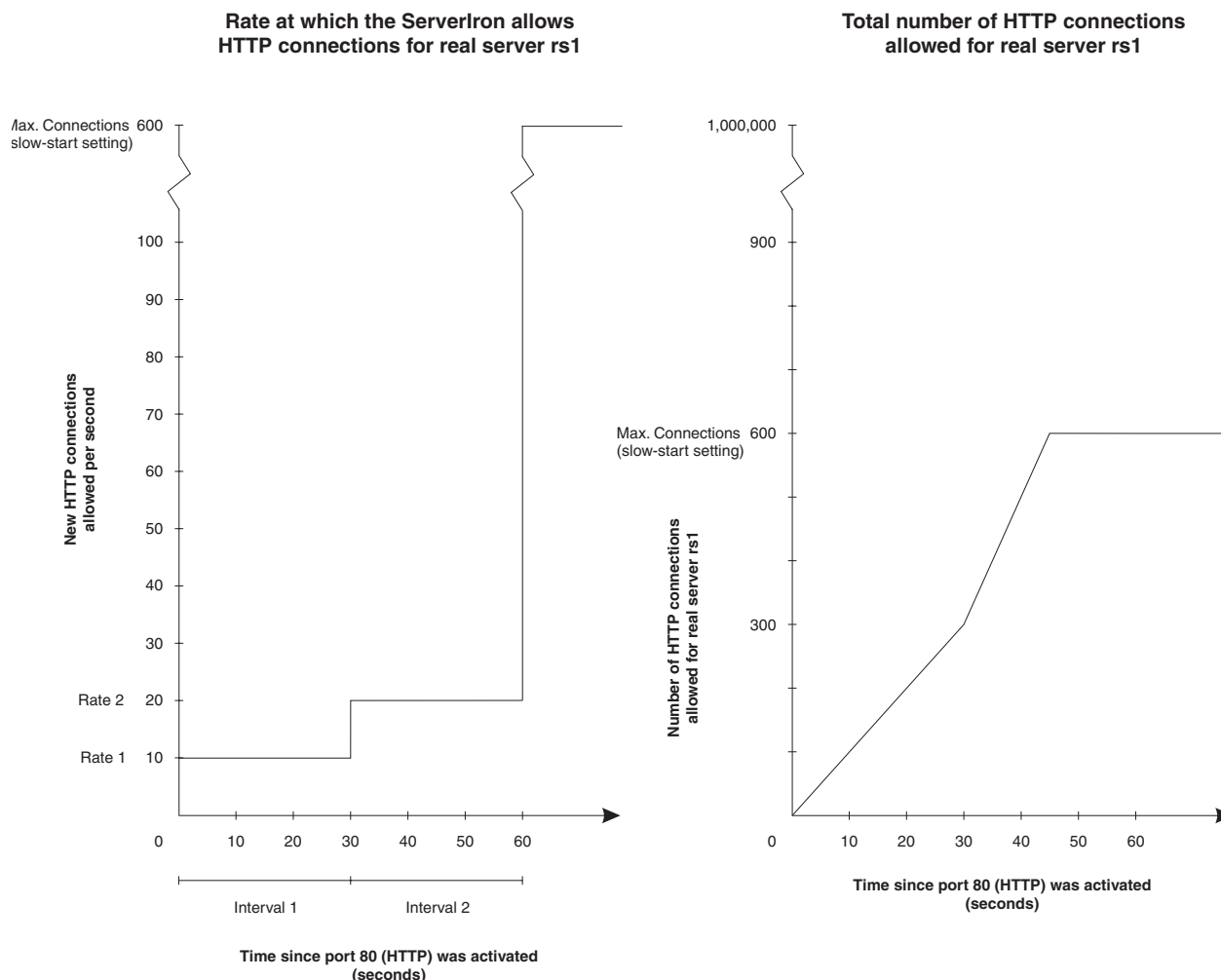
```
ServerIron(config)# server real-name rs1 192.168.1.1
ServerIron(config-rs-rs1)# port http
ServerIron(config-rs-rs1)# port http slow-start 101
ServerIron(config-rs-rs1)# exit
```

**Syntax:** port <port> slow-start <list-id>

The **port http slow-start 101** command binds slow-start list 101 (defined for port 80 above) to port 80 (HTTP) on real server rs1.

Using the slow-start list defined above, the two graphs in Figure 12.5 illustrate how a user-configured slow-start mechanism ramps up the connections for a port on a real server. The graph on the left shows the rate at which the number of HTTP connections increases over the slow-start period. The graph on the right shows how the maximum number of HTTP connections the ServerIron allows for real server rs1 increases over the slow-start period.

**Figure 12.5** Example of a user-configured slow-start mechanism for port 80 (HTTP) on a real server



The graph on the left shows the rate at which the ServerIron allows HTTP connections for real server rs1, as follows:

- From the time port 80 (HTTP) on real server rs1 is activated, until 30 seconds afterwards (until the end of interval 1), the ServerIron allows the real server up to 10 (rate 1) new HTTP connections every second.
- From 30 seconds to 60 seconds (until the end of interval 2), the ServerIron allows up to 20 (rate 2) new HTTP connections every second.
- After 60 seconds (interval 1 plus interval 2), the slow-start period ends, and the ServerIron allows up to the maximum number of connections for the server set by the <max-connections> parameter in the slow start list.

The graph on the right shows how the maximum number of possible HTTP connections for real server rs1 increases over the slow-start period:

- Ten seconds after going online, the maximum number of HTTP connections real server rs1 can have is 300: a maximum of 10 (rate 1) new HTTP connections per second for 30 (interval 1) seconds equals 300 total HTTP connections for real server rs1.
- After 30 seconds, the maximum number of HTTP connections for real server rs1 increases by 20 (rate 2) connections per second, until 600 HTTP connections (the ceiling specified by the <max-connections> parameter in the slow-start list) is reached. This ceiling of concurrent 600 HTTP connections applies for the entire time the server is active; the ServerIron allows the server no more than this number of concurrent HTTP connections.

### Applying a User-Configured Slow-Start Mechanism to Multiple Ports

To apply a user-configured slow-start mechanism to more than one port, create slow-start lists for each port and apply them to ports on one or more real servers. For example, to configure a slow-start mechanism for HTTP (port 80) and SSL (port 443):

```
ServerIron(config)# server port 80
ServerIron(config-port-80)# slow-start 100 10 30 20 30 600
ServerIron(config-port-80)# slow-start 101 20 30 40 30 1500
ServerIron(config-port-80)# exit

ServerIron(config)# server port 443
ServerIron(config-port-80)# slow-start 101 20 60 40 120 2400
ServerIron(config-port-80)# exit

ServerIron(config)# server real-name rs2 192.168.1.2
ServerIron(config-rs-rs2)# port http
ServerIron(config-rs-rs2)# port http slow-start 100
ServerIron(config-rs-rs2)# exit

ServerIron(config)# server real-name rs3 192.168.1.3
ServerIron(config-rs-rs3)# port http
ServerIron(config-rs-rs3)# port http slow-start 101
ServerIron(config-rs-rs3)# port ssl
ServerIron(config-rs-rs3)# port ssl slow-start 101
ServerIron(config-rs-rs3)# exit
```

The commands above create two slow-start lists for port 80 (HTTP) and one for port 443 (SSL). Slow-start list 100 for port 80 is applied to the HTTP port on real server rs2. Slow-start list 101 for port 80 is applied to the HTTP port on real server rs3. Slow-start list 101 for port 443 is applied to the SSL port on real server rs3. Note that slow-start list 101 for port 80 has no relation to slow-start list 101 for port 443.

In this configuration, port 80 on real server rs2 and ports 80 and 443 on real server rs3 are each subject to a user-configured slow-start mechanism. All other ports on the real servers are subject to the default slow-start mechanism described in “Default Port Slow-Start Mechanism” on page 12-64.

### Disabling the Slow Start Mechanism

You can globally disable the mechanism. When you disable the slow-start mechanism, the ServerIron can immediately send up to the maximum number of connections specified for the real server when the server comes



up. Disabling slow-start does not remove the slow-start configuration information from the real servers. To reactive slow-start, globally re-enable the feature.

To globally disable the slow-start mechanism, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# server no-slow-start
```

**Syntax:** [no] server no-slow-start

To globally re-enable slow-start, enter the following command:

```
ServerIron(config)# no server no-slow-start
```



---

# Chapter 13

## Configuring IP Forwarding

This chapter describes how to configure the ServerIron to perform IP forwarding. When you configure IP forwarding on the ServerIron, the ServerIron can perform simple IP routing to forward traffic at Layer 3 from one sub-net to another. Moreover, you can configure devices to use an IP interface on the ServerIron as the default gateway.

---

**NOTE:** This chapter applies only to the ServerIronXL, not to the ServerIron 400 or ServerIron 800. For information about Layer 3 support on the ServerIron 400 and ServerIron 800, contact Foundry Networks.

---

Using IP forwarding, you can configure the ServerIron to route IP traffic in the following configurations:

- Simple forwarding between different sub-nets within the same Layer 2 broadcast domain
- Simple forwarding between different sub-nets in different Layer 2 broadcast domains
- Forwarding from host or server to the ServerIron's management IP address when the ServerIron and the client or server are in different sub-nets
- Forwarding to a virtual IP (VIP) address when the host or server and the VIP are in different sub-nets

You can provide routing information to the ServerIron's IP route table by enabling RIP, adding static IP routes, or both.

For configuration examples, see "IP Forwarding Application Examples" on page 13-20.

---

**NOTE:** The ServerIron supports static IP routes and the Routing Information Protocol (RIP). Other routing protocols such as Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP) are not supported.

---

## Product Support

IP forwarding is supported on the ServerIronXL. The feature currently is not supported on the ServerIron 400 or ServerIron 800.

In addition, the feature is not supported on the ServerIronXL/G.

## Overview

IP forwarding enables you to configure IP interfaces on the ServerIron and use the interfaces as default gateways for your real servers and other devices.

IP forwarding provides an alternative to using source IP addresses to multinet the ServerIron. Source IP addresses allow the ServerIron to be in multiple sub-nets but the ServerIron remains a Layer 2 and Layer 4 – 7 switch.

**NOTE:** When IP forwarding is disabled, the ServerIron does not route IP traffic. The ServerIron remains a Layer 2 and Layer 4 – 7 switch.

When you enable IP forwarding on the ServerIron, the ServerIron becomes a Layer 3 IP router in addition to a Layer 2 and Layer 4 – 7 switch. You can configure multiple IP interfaces in different sub-nets to multinet the ServerIron. The ServerIron uses an IP route table to select paths for forwarding traffic.

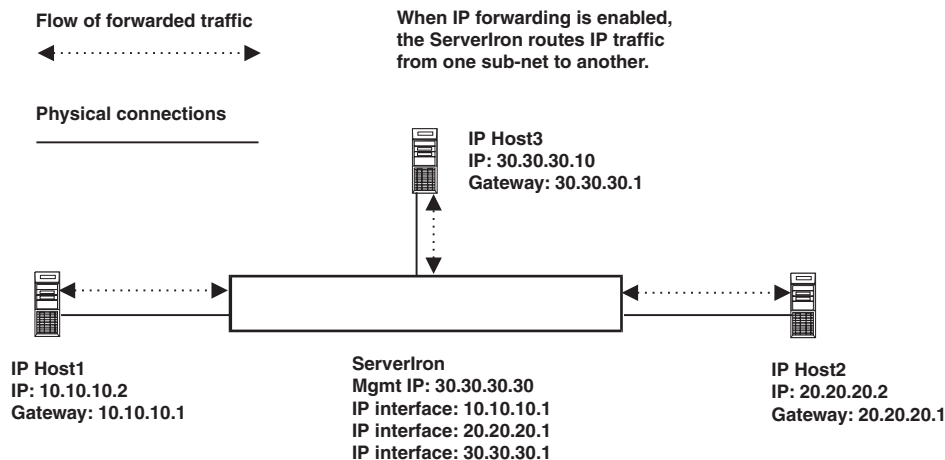
IP forwarding also allows hosts to establish management sessions with the ServerIron even when the hosts and the ServerIron's management IP address are in different sub-nets.

## Examples

Figure 13.1 shows an example of a ServerIron configured for IP forwarding.

**NOTE:** See “Forwarding Traffic From One Sub-net To Another” on page 13-22 for the CLI commands for this IP forwarding configuration.

**Figure 13.1 Forwarding IP traffic between different sub-nets in the same broadcast domain**



This example shows a ServerIron that is directly connected to three IP hosts. Each host is in a different sub-net. In addition to its management IP address, the ServerIron has three IP interfaces, one for each of the sub-nets the directly-connected IP hosts are in. When the ServerIron receives a packet from one host to another, the ServerIron looks in the IP route table for a route to destination sub-net.

- If the route table contains a forwarding entry for the destination sub-net address or host address, the ServerIron looks in the ARP cache for an entry that maps the IP address of the route's next-hop gateway to the next-hop gateway's MAC address.
- If the ARP cache contains an entry that maps the next hop IP address to its MAC address, the ServerIron changes the destination MAC address of the packet to the address in the ARP cache entry. The ServerIron then decrements the packet's Time-to-Live (TTL), and if the resulting TTL value is greater than 0, the ServerIron forwards the packet on the port(s) associated with the ARP cache entry.

However, if the resulting TTL is 0, the ServerIron drops the packet and sends an ICMP TTL Expired message back to the device that sent the packet.

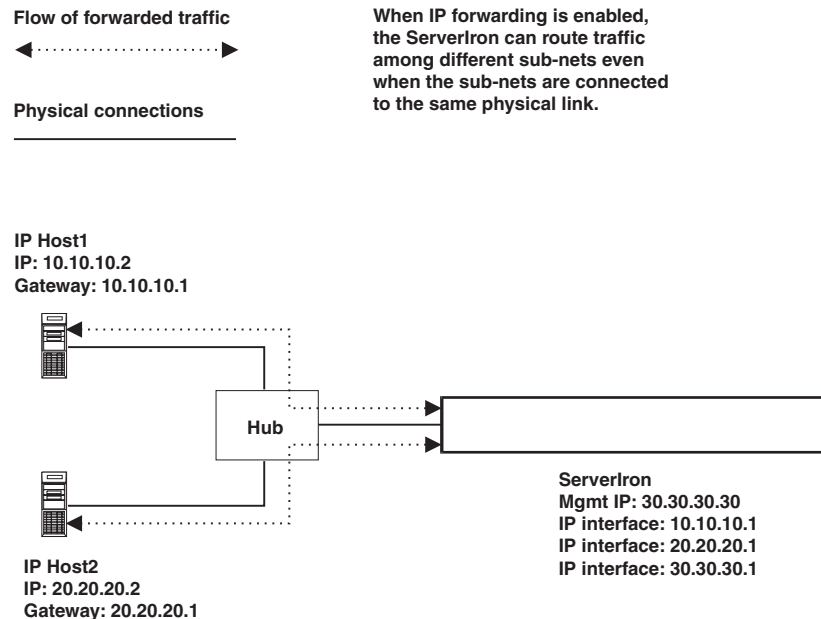
- If the ARP cache does not contain an entry for the next hop IP address, the ServerIron sends an ARP request for the address, then drops the packet.

- If the IP route table does not contain a forwarding entry, the ServerIron drops the packet and sends an ICMP Destination Network Unreachable message back to the device that sent the packet.

**NOTE:** The ServerIron has an IP interface in the same sub-net as the ServerIron's management address. This interface is required in all IP forwarding configurations, and identifies the management IP address' sub-net as a directly-connected sub-net. If you do not add the interface, the ServerIron drops packets that are addressed to a host in the management IP address' sub-net.

The ServerIron can route traffic between sub-nets even when multiple sub-nets are attached to the ServerIron through the same physical link. Figure 13.2 on page 13-3 shows an example.

**Figure 13.2 Forwarding IP traffic between different sub-nets connected to the same physical link**



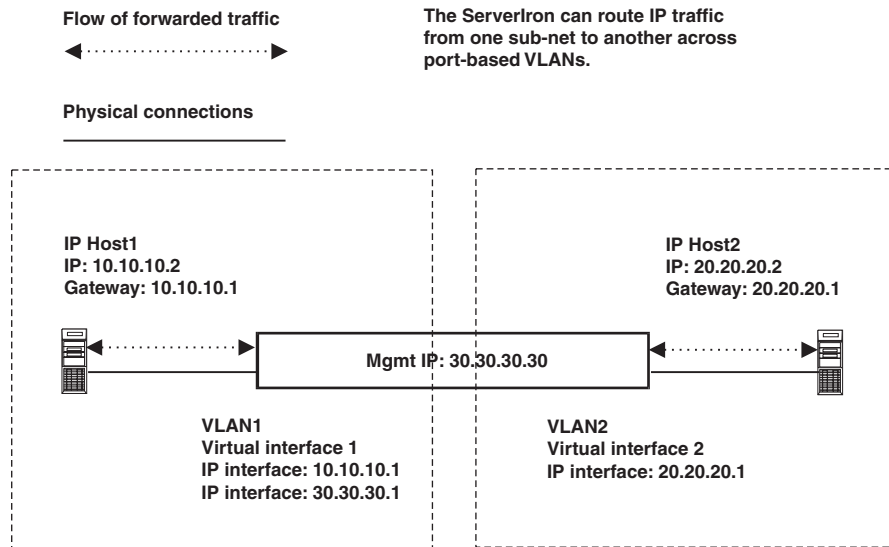
Logically, this configuration is the same as the configuration in Figure 13.1. In each case, the ServerIron routes traffic from one sub-net to the other.

The ServerIrons in Figure 13.1 and Figure 13.2 have only one port-based VLAN. However, the ServerIron can route IP traffic from one port-based VLAN to another. Figure 13.3 shows an example.

**NOTE:** Foundry recommends that you do not use VLAN 0 or VLAN 4095 when configuring additional VLANs on the device.

**NOTE:** See "Forwarding Traffic From One Sub-net To Another In Separate Broadcast Domains" on page 13-23 for the CLI commands for this IP forwarding configuration.

**Figure 13.3 Forwarding IP traffic between different sub-nets in different broadcast domains**



In this example, the ServerIron is configured with two port-based VLANs, creating two separate Layer 2 broadcast domains. The port connected to Host1 is in VLAN 1 and the port connected to Host2 is in VLAN 2. The hosts are in separate sub-nets. The ServerIron can route traffic between the sub-nets regardless of the fact that they are in separate Layer 2 broadcast domains.

## Configuring IP Forwarding

To configure IP forwarding:

- Add a virtual routing interface.
- Add an IP interface to the virtual routing interface.
- Optionally, add a static IP route.
- Optionally, enable RIP.
- Optionally, add a static entry to the ARP table. (The ServerIron automatically sends ARP requests when necessary and caches the responses.)
- Enable IP forwarding.

---

**NOTE:** Do not configure a virtual routing interface to have the same IP address as the ServerIron's management address.

---

## Virtual Routing Interfaces

IP interfaces on the ServerIron are associated with virtual routing interfaces, not with physical ports. A virtual routing interface is a logical interface associated with multiple ports that you can configure with routing parameters. A virtual interface is a logical interface that includes all the physical ports within a port-based VLAN. When you configure a virtual routing interface on the ServerIron, the interface is associated with all the ports in the VLAN that contains the virtual routing interface.

You can configure up to 24 virtual routing interfaces on the device. Each VLAN can have one virtual routing interface. Each virtual routing interface can have multiple IP interfaces.

## IP Interfaces

After you add a virtual routing interface, you can add IP interfaces to the virtual interface. Adding an IP interface places the ServerIron in the sub-net of the interface. In this sense, an IP interface provides the same function as a source IP address.

Hosts on the network can use the IP interfaces on the ServerIron as their default gateways. For example, if you add IP interface 20.20.20.1/24 to the ServerIron, servers in the 20.20.20.x sub-net can be configured to use 20.20.20.1 as their default gateway. When the ServerIron receives a packet from one of the servers, the ServerIron uses an entry in the IP route table to forward the packet.

You can configure up to 64 IP interfaces on the ServerIron. The IP interfaces can be on the same virtual routing interface or on different virtual routing interfaces. You can add more than one interface in a given sub-net.

## Management IP Address Sub-net

When IP forwarding is enabled, the ServerIron must have an IP interface in the same sub-net as the ServerIron's management IP address. When you add an interface, the software automatically creates a route to the interface's sub-net. The ServerIron requires the route for forwarding traffic to the sub-net. The software does not create a route entry when you configure the management IP address itself, so you must add an interface that is in the management IP address' sub-net.

For example, to allow a host in the 10.10.10.x sub-net to open a Telnet connection to management IP address 10.10.10.2, the ServerIron must have an IP interface in the 10.10.10.x sub-net.

You cannot use the management IP address as a default gateway address. Instead, configure IP interfaces and use those interfaces as the default gateway addresses.

---

**NOTE:** The management IP address is required. If you do not configure the management IP address, you will not be able to access the ServerIron, even through the IP interfaces configured on the virtual interface. Also, IP pings or Telnet from the ServerIron itself to some other devices will not work. However, the ServerIron will forward through traffic even if the management IP address is not configured.

---

## Default Gateway

You can specify the ServerIron's default gateway either by explicitly identifying an IP address as the default gateway (**ip default-gateway** command) or by adding a default IP route (**ip route** command). Either method is valid. In fact, the software automatically applies the address you specify with one command to the other if IP forwarding is enabled. For example, if IP forwarding is enabled and you enter the **ip default-gateway 10.10.10.1** command to specify 10.10.10.01 as the default gateway, the software automatically adds the following default route to the IP route table: 0.0.0.0 0.0.0.0 10.10.10.1.

If the IP route table already contains a default route, the software changes the route to reflect the new gateway address.

You can configure IP hosts (for example, real servers) to use the ServerIron as a default gateway by configuring an IP interface on the ServerIron that is in the same sub-net as the hosts. Do not use the ServerIron's management IP address as a default gateway.

## IP Route Table

When IP forwarding is enabled, the ServerIron uses an IP route table to route traffic from one sub-net to another. The sub-nets can be directly connected (directly reachable) or remote (reachable through a next-hop router). IP routes are associated with IP interfaces on the virtual routing interface.

Here is an example of the ServerIron's IP route table. For information about the fields in this display, see "Displaying the IP Route Table" on page 13-18.

```
Total number of IP routes: 3
Start index: 1  D:Connected  S:Static  *:Candidate default
```

	Destination	NetMask	Gateway	Port	Cost	Type
1	10.10.10.0	255.255.255.0	0.0.0.0	ve1	1	D
2	20.20.20.0	255.255.255.0	0.0.0.0	ve1	1	D
3	50.50.50.0	255.255.255.0	20.20.20.10	ve1	1	S

Route entries enter the table in the following ways:

- From RIP, if enabled.
- When you add an IP interface. The software automatically creates a directly-connected route to the sub-net the interface is in. For example, if you add IP interface 10.10.10.1/24, the software creates a route table entry for the directly-connected 10.10.10.x/24 sub-net.
- When you add a static IP route. The route table entries that the software creates when you add IP interfaces enable the ServerIron to route to those directly-connected sub-nets, but do not provide routing information to other sub-nets. To enable the ServerIron to route to a sub-net that is not directly connected to the ServerIron, add a static route for the sub-net.

The route table can contain up to 512 static IP routes. This does not include routes created when you add IP interfaces.

---

**NOTE:** The ServerIron does not support multiple paths (routes) to the same destination.

---



---

**NOTE:** The software allows you to specify a metric when you configure a static IP route. The metric is used only when RIP is enabled.

---

Although static routes do not age out, the software removes a static route if the virtual interface that has the route's IP interface goes down. A virtual interface can go down if all the physical ports in the interface go down. For example, if you add IP interface 1.1.1.1/24 to virtual interface 1, the software creates the 1.1.1.0/24 route in the IP route table. The route remains in the table indefinitely. However, if virtual interface 1 goes down, the software removes route 1.1.1.0/24 from the route table. When the interface comes back up, the software adds the route back to the IP route table.

### ARP Cache

When the ServerIron has a route for forwarding a packet to its destination, the ServerIron looks in the ARP cache for an entry that maps the route's next-hop IP address to its MAC address. The ServerIron changes the source MAC address of the packet to the ServerIron's base MAC address and changes the destination MAC address to the MAC address of the next hop. The ServerIron then decrements the TTL, and if the resulting TTL value is greater than 0, the ServerIron forwards the packet on the port(s) associated with the ARP cache entry. However, if the resulting TTL is 0, the ServerIron drops the packet and sends an ICMP TTL Expired message back to the device that sent the packet.

The ARP cache receives entries from the following sources:

- Replies to ARP requests – The ServerIron automatically creates or refreshes an ARP entry when the device receives an ARP reply. These dynamic entries age out if they are unused.
- Static entries – A static entry does not age out.

Each entry associates a port number and a VLAN ID with the IP address and MAC address. The port number identifies the port that is attached to the MAC address. The VLAN ID identifies the VLAN that the port connected to the host is in.

### ICMP Messages

If the IP route table does not have a route to a packet's destination, the ServerIron sends an ICMP Destination Network Unreachable message back to the device that sent the packet.

The ServerIron does not send an ICMP redirect message when the source (ingress) and destination (egress) ports for a packet are the same. For Layer 3 traffic (and Layer 4 – 7 traffic), this is a valid traffic occurrence for the ServerIron, as shown in Figure 13.2 on page 13-3.

---

**NOTE:** Although this is not a valid traffic occurrence for Layer 2 traffic, the ServerIron nonetheless drops the traffic, without sending a message.

---



## Supported Configurations

IP forwarding is supported in the following configurations:

- Simple forwarding between sub-nets
- Forwarding to the ServerIron management IP address
- Forwarding to a virtual IP (VIP) address configured on the ServerIron
- Server Load Balancing:
  - Load balancing Layer 4 (TCP and UDP) application port traffic
  - Load balancing Layer 7 application-specific traffic
  - Symmetric SLB (SSLB)
  - SwitchBack (direct server return)
  - Hot standby
- Web Switching:
  - URL switching
  - Cookie switching
  - Concurrent URL and cookie switching
  - HTTP header hashing
  - SSL session ID switching
- Global Server Load Balancing (GSLB)
- Transparent Cache Switching (TCS)
- Firewall load balancing (FWLB):
  - Basic Layer 3 FWLB
  - IronClad (high-availability) Layer 3 FWLB
  - Multi-zone FWLB
  - Virtual Private Network (VPN) FWLB
  - Always-active configurations
- Network Address Translation (NAT)

For configuration examples, see “IP Forwarding Application Examples” on page 13-20.

## Configuring IP Interfaces and Route Table Entries

The procedures in this section describe how to perform the following tasks:

- Add a virtual routing interface.
- Add an IP interface to the virtual routing interface.
- Optionally, add a static IP route.
- Optionally, add a static entry to the ARP table.
- Enable IP forwarding.
- Optionally, configure RIP.

---

**NOTE:** You must enable IP forwarding to place the IP forwarding configuration into effect.

---

---

**NOTE:** Do not configure a virtual routing interface to have the same IP address as the ServerIron's management address.

---

## Adding a Virtual Routing Interface

To add an IP interface to the ServerIron, you must first add a virtual routing interface. After you add the virtual routing interface, you can configure IP addresses on the routing interface.

To add a virtual interface, use the following CLI method.

### USING THE CLI

To add a virtual routing interface, enter commands such as the following:

```
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
```

The **vlan 1** command changes the CLI to the configuration level for VLAN 1. The **router-interface ve 1** command adds virtual routing interface 1.

**Syntax:** [no] router-interface ve <num>

The <num> parameter specifies the interface ID and can be from 1 – 24.

## Adding an IP Interface to the Virtual Routing Interface

After you add a virtual routing interface, you can add up to 64 IP interfaces to the virtual routing interface. To add an IP interface, use the following CLI method.

---

**NOTE:** When you add an IP interface to a virtual routing interface and the interface is up, the software adds a directly-connected static IP route to the route table for the address' sub-net. The software does not add the route unless the interface is up.

---



---

**NOTE:** For IP forwarding to work properly, you must add an IP interface that is in the same sub-net as the management IP address. This is true regardless of whether you plan to allow management access from other sub-nets.

---



---

**NOTE:** Do not configure a virtual routing interface to have the same IP address as the ServerIron's management address.

---

### USING THE CLI

To add an IP interface, enter commands such as the following:

```
ServerIron(config)# interface ve 1
ServerIron(config-vif-1)# ip address 10.10.10.1 255.255.255.0
```

The **interface ve 1** command changes the CLI to the configuration level for virtual routing interface 1. The **ip address** command adds an IP interface.

**Syntax:** [no] ip address | nat-address | standby-address <ip-addr> <ip-mask>

or

**Syntax:** [no] ip address | nat-address | standby-address <ip-addr>/<mask-bits>

The **address | nat-address | standby-address** parameter identifies the type of IP interface you are adding.

- The **address** parameter adds a standard IP interface. This option is applicable in most cases.
- The **nat-address** parameter applies to active-standby configurations. This parameter configures a shared IP interface for use with SLB source NAT. Enter the same command with the same IP address on each of the

ServerIrons in the active-standby configuration. The address is active only on one ServerIron (the ServerIron that is currently active) at a time.

---

**NOTE:** Source NAT is different from standard Network Address Translation (NAT). For information about using IP forwarding with standard NAT, see “Forwarding Traffic in a Network Address Translation Configuration” on page 13-33.

---

- The **standby-address** parameter applies to active-standby configurations and allows both ServerIrons to share the same router interface. One of the ServerIrons actively supports the interface while the other ServerIron provides failover for the interface if the first ServerIron becomes unavailable. Real servers can use the shared interface as their default gateway. Enter the same command with the same IP address on each of the ServerIrons in the active-standby configuration. The address is active only on one ServerIron (the ServerIron that is currently active) at a time.

The <ip-addr> parameter specifies the IP address.

The <ip-mask> parameter specifies a class-based (or “Classical”) IP sub-net mask.

The <mask-bits> parameter specifies the number of significant bits in a Classless Interdomain Routing (CIDR) sub-net mask.

You can use either format to configure the interface. For example, both the following commands are valid and produce the same result:

- ip address 10.10.10.1 255.255.255.0
- ip address 10.10.10.1/24

## Adding a Static IP Route

To add a static IP route to the ServerIron's IP route table, use the following CLI method.

---

**NOTE:** The software places the static route in the IP route table only if the virtual routing interface is up.

---

### USING THE CLI

To add a static IP route, enter a command such as the following at the global CONFIG level of the CLI:

```
ServerIron(config)# ip route 209.157.2.0 255.255.255.0 192.168.2.1
```

This command adds a static IP route to the 209.157.2.x/24 sub-net.

**Syntax:** [no] ip route <dest-ip-addr> <dest-mask> <next-hop-ip-addr> | null0 [<metric>]

or

**Syntax:** [no] ip route <dest-ip-addr>/<mask-bits> <next-hop-ip-addr> | null0 [<metric>]

The <dest-ip-addr> is the route's destination. The <dest-mask> is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering a forward slash followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/24. To configure a default route, enter 0.0.0.0 for <dest-ip-addr> and 0.0.0.0 for <dest-mask> (or 0 for the <mask-bits> if you specify the address in CIDR format). Specify the IP address of the default gateway using the <next-hop-ip-addr> parameter.

The <next-hop-ip-addr> is the IP address of the next-hop router (gateway) for the route. If you specify **null0** instead of a next hop IP address, the ServerIron discards packets addressed to the route's destination IP address instead of forwarding them to another device.

---

**NOTE:** If you add a default route, the gateway address of the route replaces the default gateway address configured by the **ip default-gateway** command. Likewise, if you use the **ip default-gateway** command to change the default gateway address, the gateway address in the default route is automatically changed also.

---

The <metric> parameter specifies the cost of the route and can be a number from 1 – 16. The default is 1. The metric is used by RIP. If you do not enable RIP, the metric is not used.

## Adding a Static ARP Entry

Static entries are useful in cases where you want to pre-configure an entry for a device that is not connected to the ServerIron, or you want to prevent a particular entry from aging out. The software removes a dynamic entry from the ARP cache if the ARP aging interval expires before the entry is refreshed. Static entries do not age out, regardless of whether the ServerIron receives an ARP request from the device that has the entry's address. The software places a static ARP entry into the ARP cache as soon as you create the entry.

---

**NOTE:** You can add static ARP entries regardless of whether IP forwarding is enabled.

---

To add a static ARP entry to the ServerIron, use the following CLI method.

To display the static ARP entries, see “Displaying the Static ARP Entries” on page 13-16.

### USING THE CLI

To add a static ARP entry, enter a command such as the following at the global CONFIG level of the CLI:

```
ServerIron(config)# arp 1 209.157.22.3 aaaa.bbbb.cccc ethernet 3
```

This command adds a static ARP entry that maps IP address 209.157.22.3 to MAC address aaaa.bbbb.cccc. The entry is for a MAC address connected to ServerIron port 3.

**Syntax:** [no] arp <num> <ip-addr> <mac-addr> ethernet <portnum> [vlan <vlan-id>]

The <num> parameter specifies the entry number. You can specify a number from 1 up to the maximum number of static entries allowed on the device. The ServerIronXL can have up to 64 static ARP entries by default. You can allocate more memory to increase this amount to 128 entries. See “Allocating Memory for More Static ARP Entries” on page 13-10.

The <ip-addr> command specifies the IP address of the device that has the MAC address of the entry.

The <mac-addr> parameter specifies the MAC address of the entry.

The **ethernet** <portnum> command specifies the port number attached to the device that has the MAC address of the entry.

The **vlan** <vlan-id> parameter specifies the port-based VLAN the entry belongs to. This parameter is required if the port you specify is a member of more than one port-based VLAN. Otherwise, the parameter is optional.

---

**NOTE:** The **clear arp** command clears learned ARP entries but does not remove any static ARP entries.

---

## Allocating Memory for More Static ARP Entries

The ServerIronXL can have up to 64 static ARP entries by default. To increase the maximum number of static ARP entries the device can have, enter a command such as the following at the global CONFIG level of the CLI:

```
ServerIron(config)# system-max ip-static-arp 128
```

**Syntax:** [no] system-max ip-static-arp <num>

The <num> parameter indicates the maximum number of static ARP entries and can be from 64 – 128. The default is 64.

## Enabling IP Forwarding

IP forwarding parameters, including IP interfaces and route table entries, do not take effect until you enable IP forwarding. IP forwarding is disabled by default. When the feature is disabled, the ServerIron is a Layer 2 and Layer 4 – 7 switch and does not perform Layer 3 routing. You can enable IP forwarding at any time.

To enable IP forwarding, use the following CLI method.

### USING THE CLI

To enable IP forwarding, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# ip forward
```

**Syntax:** [no] ip forward

## Configuring RIP

RIP is disabled by default. If you want the ServerIron to use RIP you must enable the protocol globally, then enable RIP on the virtual routing interface and specify the version (version 1 only, version 2 only, or version 1 compatible with version 2).

Optionally, you also can set or change the following parameters:

- Route redistribution – You can enable the software to redistribute static routes from the IP route table into RIP. Redistribution is disabled by default.
- Learning of default routes – The default is disabled.
- Loop prevention (split horizon or poison reverse) – The default is poison reverse.

### Enabling RIP

RIP is disabled by default. To enable it, use the following CLI method. You must enable the protocol both globally and on the virtual routing interface.

#### USING THE CLI

To enable RIP globally, enter the following command:

```
ServerIron(config)# router rip
```

**Syntax:** [no] router rip

To enable RIP on the virtual routing interface and specify the RIP version, enter commands such as the following:

```
ServerIron(config-rip-router)# interface ve 1  
ServerIron(config-vif-1)# ip rip v1-only
```

This command changes the CLI to the configuration level for virtual routing interface 1 and enables RIP version 1 on the interface. You must specify the version.

**Syntax:** interface ve <num>

**Syntax:** [no] ip rip v1-only | v1-compatible-v2 | v2-only

### Enabling Redistribution of IP Static Routes into RIP

By default, the software does not redistribute the IP static routes in the route table into RIP. To configure redistribution, perform the following tasks:

- Configure redistribution filters (optional). You can configure filters to permit or deny redistribution for a route based on the route's metric. You also can configure a filter to change the metric. You can configure up to 64 redistribution filters. The software uses the filters in ascending numerical order and immediately takes the action specified by the filter. Thus, if filter 1 denies redistribution of a given route, the software does not redistribute the route, regardless of whether a filter with a higher ID permits redistribution of that route.

---

**NOTE:** The default redistribution action is permit, even after you configure and apply a permit or deny filter. To deny redistribution of specific routes, you must configure a deny filter.

---

---

**NOTE:** The option to set the metric is not applicable to static routes.

---

- Enable redistribution.

---

**NOTE:** If you plan to configure redistribution filters, do not enable redistribution until you have configured the filters.

---

### USING THE CLI

When you enable redistribution, all IP static routes are redistributed by default. If you want to deny certain routes from being redistributed into RIP, configure deny filters for those routes before you enable redistribution. You can configure up to 64 RIP redistribution filters. They are applied in ascending numerical order.

---

**NOTE:** The default redistribution action is still permit, even after you configure and apply redistribution filters to the virtual routing interface. If you want to tightly control redistribution, apply a filter to deny all routes as the last filter (filter ID 64), then apply filters with lower filter IDs to allow specific routes.

---

To configure a redistribution filter, enter a command such as the following:

```
ServerIron(config-rip-router)# deny redistribute 1 static address 207.92.0.0
255.255.0.0
```

This command denies redistribution of all 207.92.x.x IP static routes.

**Syntax:** [no] permit | deny redistribute <filter-num> static address <ip-addr> <ip-mask>  
[match-metric <value> | set-metric <value>]

The <filter-num> specifies the redistribution filter ID. Specify a number from 1 – 64. The software uses the filters in ascending numerical order. Thus, if filter 1 denies a route from being redistributed, the software does not redistribute that route even if a filter with a higher ID permits redistribution of the route.

The **address** <ip-addr> <ip-mask> parameters apply redistribution to the specified network and sub-net address. Use 0 to specify “any”. For example, “207.92.0.0 255.255.0.0” means “any 207.92.x.x sub-net”. However, to specify any sub-net (all sub-nets match the filter), enter “address 255.255.255.255 255.255.255.255”.

The **match-metric** <value> parameter applies redistribution to those routes with a specific metric value; possible values are from 1 – 15.

The **set-metric** <value> parameter sets the RIP metric value that will be applied to the routes imported into RIP.

---

**NOTE:** The **set-metric** parameter does not apply to static routes.

---

The following command denies redistribution of a 207.92.x.x IP static route only if the route’s metric is 5.

```
ServerIron(config-rip-router)# deny redistribute 2 static address 207.92.0.0
255.255.0.0 match-metric 5
```

The following commands deny redistribution of all routes except routes for 10.10.10.x and 20.20.20.x:

```
ServerIron(config-rip-router)# deny redistribute 64 static address 255.255.255.255
255.255.255.255
ServerIron(config-rip-router)# permit redistribute 1 static address 10.10.10.0
255.255.255.0
ServerIron(config-rip-router)# permit redistribute 2 static address 20.20.20.0
255.255.255.0
```

### Enabling Redistribution

After you configure redistribution parameters, you need to enable redistribution.

### USING THE CLI

To enable RIP redistribution, enter the following command:

```
ServerIron(config-rip-router)# redistribution
```

**Syntax:** [no] redistribution

## Enabling Learning of Default Routes

By default, the software does not learn RIP default routes. You can enable learning of RIP default routes using the following CLI method.

### *USING THE CLI*

To enable learning of default RIP routes, enter commands such as the following:

```
ServerIron(config)# interface ve 1
ServerIron(config-vif-1)# ip rip learn-default
```

**Syntax:** interface ve <num>

**Syntax:** [no] ip rip learn-default

## Changing the Route Loop Prevention Method

RIP can use the following methods to prevent routing loops:

- Split horizon – The ServerIron does not advertise a route on the same interface as the one on which the ServerIron learned the route.
- Poison reverse – The ServerIron assigns a cost of 16 (“infinite” or “unreachable”) to a route before advertising it on the same interface as the one on which the ServerIron learned the route. This is the default.

---

**NOTE:** These methods are in addition to RIP’s maximum valid route cost of 15.

---

### *USING THE CLI*

To enable split horizon, enter commands such as the following:

```
ServerIron(config)# interface ve 1
ServerIron(config-vif-1)# no ip rip poison-reverse
```

**Syntax:** [no] ip rip poison-reverse

## Converting an Existing Configuration to Work with IP Forwarding

Software releases with IP forwarding fully support ServerIron configurations created using earlier software releases. You do not need to completely reconfigure the ServerIron to use IP forwarding. You only need to add the IP forwarding information and make minor modifications to existing information.

To convert a configuration that you created for the ServerIron as a Layer 2 and Layer 4 – 7 switch into a configuration that supports IP forwarding, perform the following steps:

- Remove the source IP addresses, if any.
- Configure virtual routing interfaces, and add an IP interface to replace each of the source IP addresses you removed from the configuration.
- Optionally, add static IP routes for remote (not directly-attached) sub-nets, if applicable.
- Optionally, enable and configure RIP.
- Optionally, change the default gateways on hosts attached to the ServerIron to one of the IP interfaces you added to the ServerIron. Doing this makes the ServerIron the default gateway for the hosts.

By adding IP interfaces that have the same addresses as the source IP addresses, you can eliminate the need to reconfigure other devices in the network that use the source IP addresses as default gateways.

## Displaying IP Forwarding Configuration Information and Statistics

You can display the following IP forwarding information:

- The IP forwarding state (enabled or disabled)

- ARP entries
- IP interfaces
- The IP route table
- IP traffic statistics

## Displaying the IP Forwarding State

To display the state (enabled or disabled) of the IP forwarding feature, use the following CLI method.

### USING THE CLI

To display IP forwarding state information as well as other global IP parameters, enter the following command at any level of the CLI:

```
ServerIron(config)# show ip

      Enabled : IP_Forwarding

      Disabled : RIP  RIP-Redist

Switch IP address: 192.168.2.100

      Subnet mask: 255.255.255.0

Default router address: 192.168.2.1
      TFTP server address: None
Configuration filename: None
      Image filename: None
```

**Syntax:** show ip

This display shows the following information.

**Table 13.1: CLI Display of Global IP Configuration Information**

This Field...	Displays...
<b>IP configuration</b>	
IP Forwarding state	<p>The state of the IP forwarding feature. The state can be one of the following:</p> <ul style="list-style-type: none"> <li>• Disabled</li> <li>• Enabled</li> </ul>
RIP state	<p>The state of RIP. The state can be one of the following:</p> <ul style="list-style-type: none"> <li>• Disabled</li> <li>• Enabled</li> </ul> <p>If route redistribution is enabled, "RIP -Redist" is displayed as well. For information, see "Enabling Redistribution of IP Static Routes into RIP" on page 13-11.</p>
Switch IP address	The management IP address you configured on the ServerIron. Specify this address for Telnet or Web management access.
Subnet mask	The sub-net mask for the management IP address.



Table 13.1: CLI Display of Global IP Configuration Information (Continued)

This Field...	Displays...
Default router address	The address of the default gateway, if you specified one.  <b>Note:</b> When IP forwarding is enabled, the address is listed only if the corresponding virtual interface is up. When IP forwarding is disabled, the configured default gateway address is always displayed.
<b>Most recent TFTP access</b>	
TFTP server address	The IP address of the most-recently contacted TFTP server, if the ServerIron has contacted a TFTP server since the last time the software was reloaded or the ServerIron was rebooted.
Configuration filename	The name under which the ServerIron's startup-config file was uploaded or downloaded during the most recent TFTP access.
Image filename	The name of the ServerIron flash image (system software file) that was uploaded or downloaded during the most recent TFTP access.

## Displaying ARP Entries

To display ARP entries, use the following CLI methods. You can display the ARP cache or the static ARP table. The ARP table contains the static ARP entries, if any, you configured on the device. The ARP cache contains all the ARP entries, including static entries.

### Displaying the ARP Cache

#### USING THE CLI

To display the ARP cache, enter the following command at any level of the CLI:

```
ServerIron(config)# show arp
```

IP	Mac	Type	Port	Age	VlanId
10.10.10.10	00d0.0958.9b07	Static	9	0	1
192.168.2.14	0050.04bb.81fa	Static	15	0	1
192.168.2.1	00e0.5205.9056	Static	15	0	1
192.168.2.157	00e0.2972.2ab5	Dynamic	15	0	1
192.168.2.15	0010.5ad1.3701	Dynamic	15	0	1
192.168.2.77	00e0.5202.de72	Dynamic	15	0	1

Total Arp Entries : 6

**Syntax:** show arp [<ip-addr> [<ip-mask>] | ethernet <portnum> mac-address <xxxx.xxxx.xxxx> [<mask>]]

The <ip-addr> and <ip-mask> parameters let you restrict the display to entries for a specific IP address and network mask. Specify the IP address masks in standard decimal mask format (for example, 255.255.0.0).

**NOTE:** The <ip-mask> parameter and <mask> parameter perform different operations. The <ip-mask> parameter specifies the network mask for a specific IP address, whereas the <mask> parameter provides a filter for displaying multiple MAC addresses that have specific values in common.

Specify the MAC address mask as “f”s and “0”s, where “f”s are significant bits. Specify IP address masks in standard decimal mask format (for example, 255.255.0.0).

The **ethernet** <portnum> parameter lets you restrict the display to entries for a specific port.

The **mac-address** <xxxx.xxxx.xxxx> parameter lets you restrict the display to entries for a specific MAC address.

The <mask> parameter lets you specify a mask for the **mac-address** <xxxx.xxxx.xxxx> parameter, to display entries for multiple MAC addresses. Specify the MAC address mask as “f”s and “0”s, where “f”s are significant bits.

Here are some examples of how to use these commands.

The following command displays all ARP entries for MAC addresses that begin with “abcd”:

```
ServerIron# show arp mac-address a.b.c.d ffff.0000.0000
```

The following command displays all IP address entries for IP addresses that begin with “209.157”:

```
ServerIron# show arp 209.157.0.0 255.255.0.0
```

This **show arp** command displays the following information.

**Table 13.2: CLI Display of ARP Cache**

This Field...	Displays...
IP	The IP address of the device.
MAC	The MAC address of the device.
Type	The type, which can be one of the following: <ul style="list-style-type: none"> <li>Dynamic – The ServerIron learned the entry from an incoming packet.</li> <li>Static – You added the entry to the ARP table.</li> </ul>
Port	The port on which the entry was learned.
Age	The number of minutes the entry has remained unused. If this value reaches the ARP aging period, the entry is removed from the table. <b>Note:</b> Static entries do not age out.
VlanId	The port-based VLAN that the ServerIron port connected to the entry’s MAC address is in.
Total ARP Entries	The total number of entries in the cache. The total includes both dynamic (learned) and static ARP entries.

## Displaying the Static ARP Entries

### USING THE CLI

To display static ARP entries, enter the following command at any level of the CLI:

```
ServerIron(config)# show ip static-arp
Static ARP table size: 64, configurable from 64 to 128
  Index   IP Address      MAC Address      Port
  1       10.10.10.10     00d0.0958.9b07   9
  2       192.168.2.1     00e0.5205.9056   15
  3       192.168.2.157   00e0.2972.2ab5   15
  4       192.168.2.14    0050.04bb.81fa   15
  5       192.168.2.15    0010.5ad1.3701   15
```

**Syntax:** show ip static-arp [<ip-addr> [<ip-mask>] | ethernet <portnum> mac-address <xxxx.xxxx.xxxx> [<mask>]]

The parameters are the same as those for the **show arp** command.

The **show ip static-arp** command displays the following information.

**Table 13.3: CLI Display of Static ARP Table**

This Field...	Displays...
Static ARP table size	The maximum number of static entries that can be configured on the device using the current memory allocation. The range of valid memory allocations for static ARP entries is listed after the current allocation. To change the memory allocation for static ARP entries, see "Allocating Memory for More Static ARP Entries" on page 13-10.
Index	The number of this entry in the table. You specify the entry number when you create the entry.
IP Address	The IP address of the device.
MAC Address	The MAC address of the device.
Port	The port attached to the device the entry is for.

## Displaying IP Interfaces

To display a list of all the IP interfaces configured on the ServerIron, use the following CLI method.

### USING THE CLI

To display a list of the IP interfaces configured on the ServerIron, enter the following command at any level of the CLI:

```
ServerIron(config)# show ip interface
```

Interface	IP-Address	OK?	Method	Status	Protocol
Ve 1	192.168.2.1	YES	manual	up	up
Ve 1	10.10.10.1	YES	manual	up	up
Ve 1	20.20.20.1	YES	manual	up	up
Ve 10	120.120.120.1	YES	manual	down	up
Ve 10	130.130.130.1	YES	manual	down	up

**Syntax:** show ip interface

This command displays the following information.

**Table 13.4: CLI Display of IP Interfaces**

This Field...	Displays...
Interface	The virtual routing interface.
IP-Address	The IP address of the interface.
OK?	Whether the IP address has been configured on the interface.

**Table 13.4: CLI Display of IP Interfaces (Continued)**

This Field...	Displays...
Method	Whether the IP address has been saved in NVRAM. If you have set the IP address for the interface in the CLI or Web Management interface, but have not saved the configuration, the entry for the interface in the Method field is "manual".
Status	The link status of the interface. The status can be one of the following: <ul style="list-style-type: none"> <li>• down</li> <li>• up</li> </ul>
Protocol	Whether the interface can provide two-way communication. If the IP address is configured, and the link status of the interface is up, the entry in the protocol field is "up". Otherwise the entry in the protocol field is "down".

## Displaying the IP Route Table

To display the IP route table, use the following CLI method.

### USING THE CLI

To display the IP route table, enter the following command at any level of the CLI:

```
ServerIron(config)# show ip route
Total number of IP routes: 9
Start index: 1 D:Connected S:Static *:Candidate default
```

	Destination	NetMask	Gateway	Port	Cost	Type
1	10.10.10.0	255.255.255.0	0.0.0.0	ve1	1	D
2	20.20.20.0	255.255.255.0	0.0.0.0	ve1	1	D
3	50.50.50.0	255.255.255.0	20.20.20.10	ve1	1	S
4	60.60.60.0	255.255.255.0	20.20.20.10	ve1	1	S
5	70.70.70.0	255.255.255.0	120.120.120.10	ve1	1	S
6	120.120.120.0	255.255.255.0	0.0.0.0	ve1	1	D
7	130.130.130.0	255.255.255.0	0.0.0.0	ve1	1	D
8	192.168.2.0	255.255.255.0	0.0.0.0	ve1	1	D
9	0.0.0.0	0.0.0.0	192.168.2.1	ve1	1	S

**Syntax:** show ip route

This command displays the following information.

**Table 13.5: CLI Display of IP Route Table**

This Field...	Displays...
Total number of IP routes	The total number of routes in the table, including routes that you added and directly-connected routes the software added when you added IP interfaces.
Start index	The starting entry number in the table.

Table 13.5: CLI Display of IP Route Table (Continued)

This Field...	Displays...
Destination	The destination network of the route.
NetMask	The network mask of the destination address.
Gateway	The next-hop router.
Port	The virtual routing interface to which the route belongs.
Cost	The route's cost.
Type	The route type, which can be one of the following: <ul style="list-style-type: none"> <li>• D – The destination is directly connected to the ServerIron.</li> <li>• R – The route is a RIP route.</li> <li>• S – The route is a static route.</li> </ul>

## Displaying IP Traffic Statistics

To display IP forwarding traffic statistics, use the following CLI method.

### USING THE CLI

To display the IP route table, enter the following command at any level of the CLI:

```
ServerIron(config)# show ip traffic
IP Statistics
  587 received, 593 sent, 14 forwarded
  0 fragmented, 0 reassembled, 0 bad header
  489 no route, 0 unknown proto, 0 no buffer, 9 other errors
```

*<the other sections in this display do not apply to IP forwarding and are not shown in this example>*

**Syntax:** show ip traffic

This command displays the following information related to IP forwarding.

Table 13.6: CLI Display of IP Forwarding Traffic Statistics

This Field...	Displays...
received	The total number of IP packets received by the device.
sent	The total number of IP packets originated and sent by the device.
forwarded	The total number of IP packets received by the device and forwarded to other devices.
filtered	The total number of IP packets filtered by the device.
fragmented	The total number of IP packets fragmented by this device to accommodate the MTU of this device or of another device.
reassembled	The total number of fragmented IP packets that this device re-assembled.

**Table 13.6: CLI Display of IP Forwarding Traffic Statistics (Continued)**

This Field...	Displays...
bad header	The number of IP packets dropped by the device due to a bad packet header.
no route	The number of packets dropped by the device because there was no route.
unknown proto	The number of packets dropped by the device because the value in the Protocol field of the packet header is unrecognized by this device.
no buffer	This information is used by Foundry customer support.
other errors	The number of packets that this device dropped due to error types other than the types listed above.

The display contains additional sections of statistics. However, the additional statistics apply to Layer 4 – 7 switching, not to IP forwarding.

## IP Forwarding Application Examples

The examples in this section illustrate implementations of the IP forwarding feature. Examples are provided for single and multiple Layer 2 broadcast domains.

**Table 13.7: IP Forwarding Application Examples**

Application	See page...
Using the ServerIron as default gateway	13-20
Forwarding between devices on different sub-nets	13-22
Forwarding management access to the ServerIron from a different sub-net	13-23
Forwarding traffic between a client and VIP on different sub-nets	13-24
Forwarding Symmetric SLB (active-active) traffic between sub-nets	13-26
Forwarding SwitchBack (direct server return) traffic between sub-nets	13-28
Forwarding traffic in an SLB hot-standby (active-standby) configuration	13-30
Forwarding traffic in an Transparent Cache Switching (TCS) configuration	13-32
Forwarding traffic in a Network Address Translation (NAT) configuration	13-33
Forwarding traffic in a Global SLB (GSLB) configuration	13-33
Forwarding traffic in a high-availability (active-standby) FWLB configuration	13-35

### Using the ServerIron as a Default Gateway

IP forwarding allows you to use an IP interface on the ServerIron itself as the default gateway for other devices in the network. For example, you can configure your real servers to use the ServerIron as their default gateway to other sub-nets.

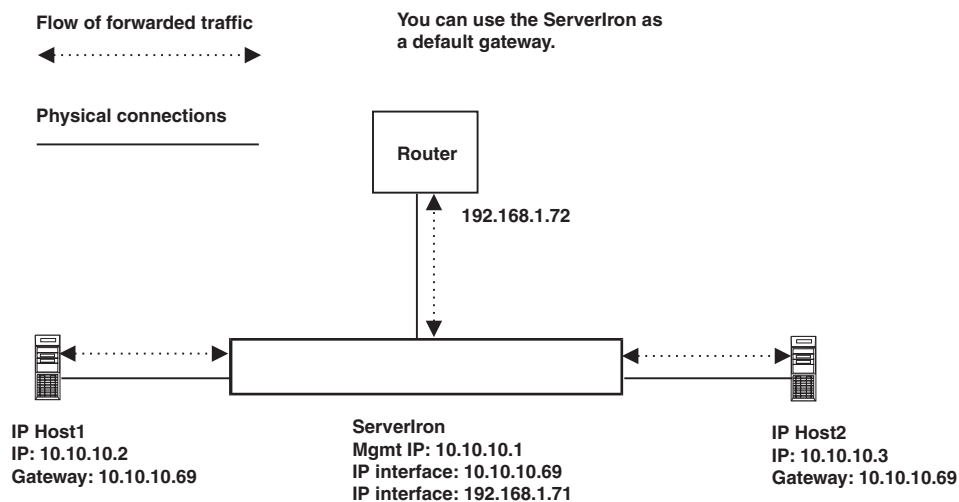
Figure 13.4 on page 13-21 shows an example of two real servers that use the ServerIron as their default gateway. Notice that the real servers are not using the ServerIron's management IP address as the gateway. You cannot use the management IP address as the gateway. Instead, configure an IP interface in the same sub-net as the real servers and specify the IP interface as the default gateway on the real servers.

The ServerIron also needs a way to reach its gateway. You can accomplish this by doing either of the following:

- Explicitly specifying the default gateway address (**ip default-gateway** command)
- Adding a default route to the IP route table (**ip route** command)

**NOTE:** For simplicity, the hosts and the ServerIron's management IP address in this example are in the same sub-net. You also can use the ServerIron as a default gateway when the hosts are in other sub-nets. In either case, add an IP interface in the host's sub-net to the ServerIron, and configure the host server to use that address as its default gateway.

**Figure 13.4 Real servers using ServerIron as their default gateway**



Here are the CLI commands for implementing the IP forwarding configuration shown in this example:

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 10.10.10.1
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip address 10.10.10.69 255.255.255.0
ServerIron(config-vif-1)# ip address 192.168.1.71 255.255.255.0
ServerIron(config-vif-1)# exit
ServerIron(config)# ip route 0.0.0.0 0.0.0.0 192.168.1.72
ServerIron(config)# ip forward
```

The first **ip address** command, at the global CONFIG level, adds the management IP address. The **ip address** commands at the virtual interface configuration level add the IP interfaces that place the ServerIron in the 10.10.10.x and 192.168.1.x sub-nets. One of the interfaces serves as the default gateway interface for the real servers. The other interface places the ServerIron in the router's sub-net. The **ip route** command specifies the ServerIron's default gateway.

## Forwarding Traffic From One Sub-net To Another

To configure the ServerIron to forward IP traffic at Layer 3 from one sub-net to another, perform the following steps. For each directly-connected sub-net, including the sub-net the ServerIron's management IP address is in, do the following:

- Configure a virtual interface.
- Change to the configuration level for the new virtual interface.
- Add an IP interface that is in the directly-connected sub-net. You also must add an IP interface that is in the same sub-net as the ServerIron's management IP address. This is required since adding the interface also causes the software to create a route to the management IP address' sub-net.
- Optionally, on each client that is on a sub-net directly connected to the ServerIron, set the client's default gateway to the IP interface on the ServerIron that is in the same sub-net as the client. If the client is in the same sub-net as the ServerIron's management IP address, specify the IP interface that you added.

For each remote sub-net, do one or both of the following:

- Enable RIP.
- Add a static IP route for the sub-net or configure a default route for all the remote sub-nets.

---

**NOTE:** You can add a default route even if you also add routes for specific sub-nets.

---

Figure 13.1 on page 13-2 shows an example of a simple configuration in which IP forwarding is used. The following commands show how to implement IP forwarding on the ServerIron in Figure 13.1. The commands also apply to the configuration in Figure 13.3 on page 13-4.

The following commands configure the ServerIron's management IP address and default gateway. (Figure 13.1 and Figure 13.3 do not show the device the ServerIron is using for its default gateway.)

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 30.30.30.30
ServerIron(config)# ip default-gateway 40.40.40.1
```

The following commands change the CLI to the configuration level for VLAN 1, then add a virtual routing interface to the VLAN. Make sure you create the virtual routing interface on the VLAN that contains the ports that connect the ServerIron to the other sub-nets.

```
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
```

### Configuring Forwarding for Directly-Connected Sub-Nets

The following commands add IP interfaces to the virtual routing interface. Each address gives the ServerIron an interface in a different sub-net. Notice that an IP interface also is added for the sub-net that contains the management IP address. This interface is required, since adding the interface also causes the software to create a directly-connected IP route to the sub-net. Add an IP interface for each directly-connected sub-net.

```
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip address 10.10.10.1 255.255.255.0
ServerIron(config-vif-1)# ip address 20.20.20.1 255.255.255.0
ServerIron(config-vif-1)# ip address 30.30.30.1 255.255.255.0
ServerIron(config-vif-1)# exit
```

### Configuring Forwarding for Remote Sub-Nets

The following commands add static IP routes that allow the ServerIron to route to remote sub-nets. The ServerIron does not have an interface in the remote sub-nets. (Figure 13.1 and Figure 13.3 do not show the remote sub-nets.)

```
ServerIron(config)# ip route 209.157.2.0 255.255.255.0 192.168.2.1
```



## Enabling RIP

The following command enables RIP version 2.

```
ServerIron(config)# router rip
```

## Enabling IP Forwarding

The following command enables IP forwarding. The IP interfaces configured above do not take effect until you enable IP forwarding.

```
ServerIron(config)# ip forward
```

## Forwarding Traffic From One Sub-net To Another In Separate Broadcast Domains

As shown in Figure 13.3 on page 13-4, the ServerIron can route IP traffic from one sub-net to another even when the sub-nets are in different port-based VLANs. (Each port-based VLAN is a separate Layer 2 broadcast domain.) To configure IP forwarding on the ServerIron in Figure 13.3, enter the following commands.

The following commands change the CLI to the global CONFIG level and add the ServerIron's management IP address.

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 30.30.30.30
```

The following commands add a virtual routing interface to VLAN 1, then add two IP interfaces to the virtual routing interface. The first IP interface allows the ServerIron to route to the sub-net that contains the management IP address. The second IP interface places the ports in VLAN 1 in Host1's sub-net.

```
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip address 30.30.30.1 255.255.255.0
ServerIron(config-vif-1)# ip address 10.10.10.1 255.255.255.0
```

The following commands add port-based VLAN 2, identify the ports in the VLAN, and add a virtual routing interface and IP interfaces.

The **untagged** command identifies the ports in the VLAN. Since VLAN 1 is the default VLAN, it automatically contains all the ServerIron ports that are not members of other VLANs. Therefore, you do not need to identify the ports in VLAN 1. VLAN 2 is a new VLAN, so you must identify the ports in the VLAN. The software removes the ports you specify from VLAN 1 (the default VLAN) and places them in VLAN 2.

The IP interface places VLAN 2's ports in Host2's sub-net.

```
ServerIron(config)# vlan 2
ServerIron(config-vlan-2)# untagged ethernet 13 to 24
ServerIron(config-vlan-2)# router-interface ve 2
ServerIron(config-vlan-2)# interface ve 2
ServerIron(config-vif-2)# ip address 20.20.20.1 255.255.255.0
ServerIron(config-vif-2)# exit
```

The following command enables IP forwarding.

```
ServerIron(config)# ip forward
```

## Forwarding Management Access from a Different Sub-net

To configure a ServerIron to allow access to its management IP address from other sub-nets, add an IP interface that is in the same sub-net as the management IP address. When you add the interface, the software creates a directly-connected route to the management IP address' sub-net.

In Figure 13.1 and Figure 13.3, the management IP address is 30.30.30.30. To allow access from other sub-nets to the sub-net that contains the ServerIron's management IP address, enter the following commands.

```
ServerIron(config)# vlan 1
```

```
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip address 30.30.30.1 255.255.255.0
```

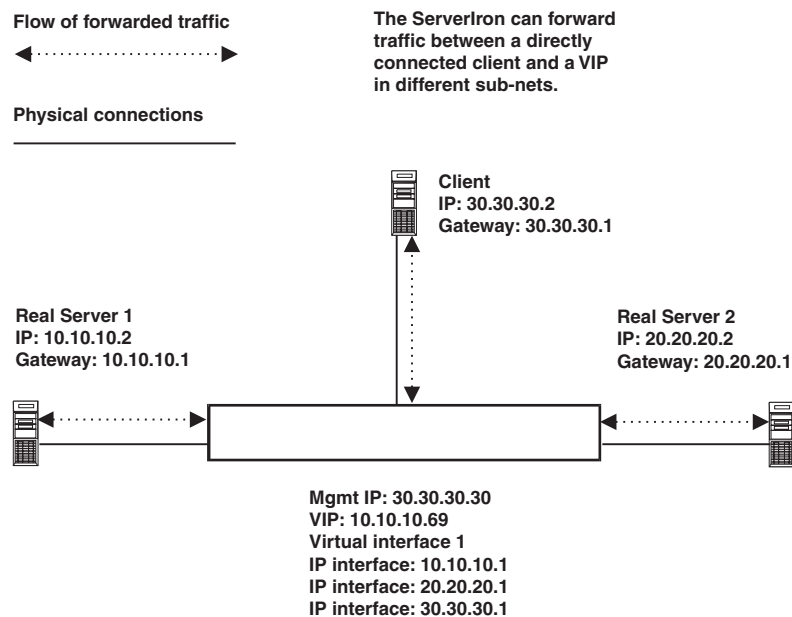
**NOTE:** For IP forwarding to work properly, you must add an IP interface that is in the same sub-net as the management IP address. This is true regardless of whether you plan to allow management access from other sub-nets.

## Forwarding Traffic Between a Client and VIP on Different Sub-nets

IP forwarding allows you to configure a VIP and its clients in different sub-nets. In addition, IP forwarding allows the VIP and the ServerIron's management IP address to be in different sub-nets. Without IP forwarding, the VIP must be in the same sub-net as the client and the management IP address.

Figure 13.5 shows an example of a configuration in which a VIP and a directly connected client are in different sub-nets.

**Figure 13.5** VIP and real servers in different sub-nets



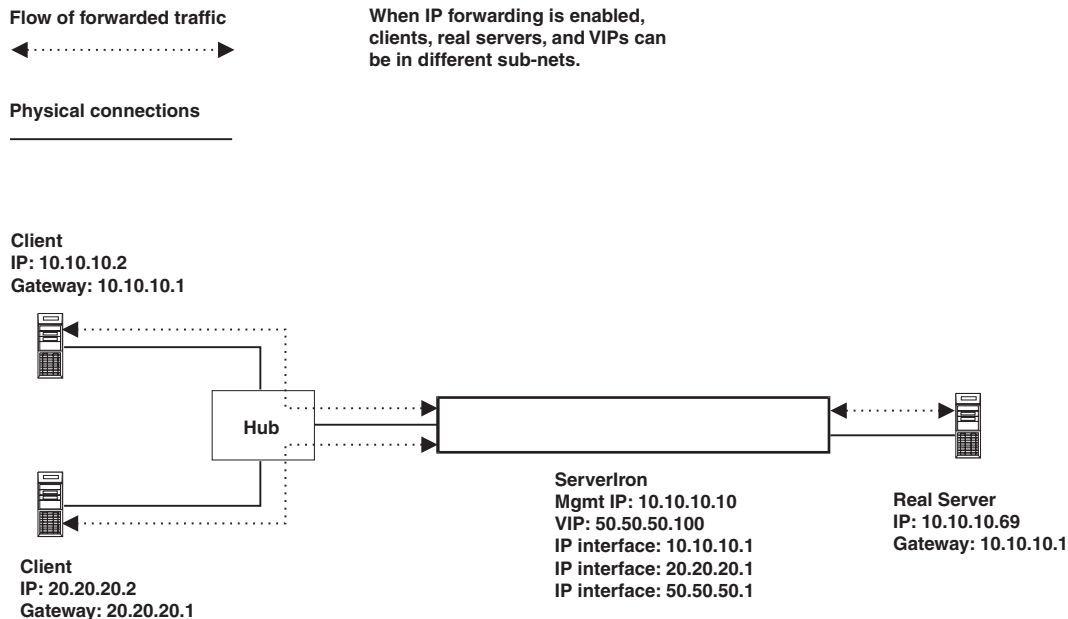
To configure IP forwarding on the ServerIron in Figure 13.5, enter the following commands. The procedures and commands for configuring SLB or Web Switching are the same with or without IP forwarding, so the SLB commands are not shown in this example.

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 30.30.30.30
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip address 10.10.10.1 255.255.255.0
ServerIron(config-vif-1)# ip address 20.20.20.1 255.255.255.0
ServerIron(config-vif-1)# ip address 30.30.30.1 255.255.255.0
ServerIron(config-vif-1)# exit
ServerIron(config)# ip forward
```

Figure 13.5 shows how to configure IP forwarding for SLB, but you also can use this type of IP forwarding configuration for Web Switching.

Figure 13.6 on page 13-25 shows another example of a configuration in which directly connected clients and a VIP are in different sub-nets. In this example, multiple sub-nets use the same physical link on the ServerIron.

**Figure 13.6 VIP and real servers in different sub-nets**



To configure IP forwarding on the ServerIron in Figure 13.5, enter the following commands.

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 10.10.10.10
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip address 10.10.10.1 255.255.255.0
ServerIron(config-vif-1)# ip address 20.20.20.1 255.255.255.0
ServerIron(config-vif-1)# ip address 50.50.50.1 255.255.255.0
ServerIron(config-vif-1)# exit
ServerIron(config)# ip forward
```

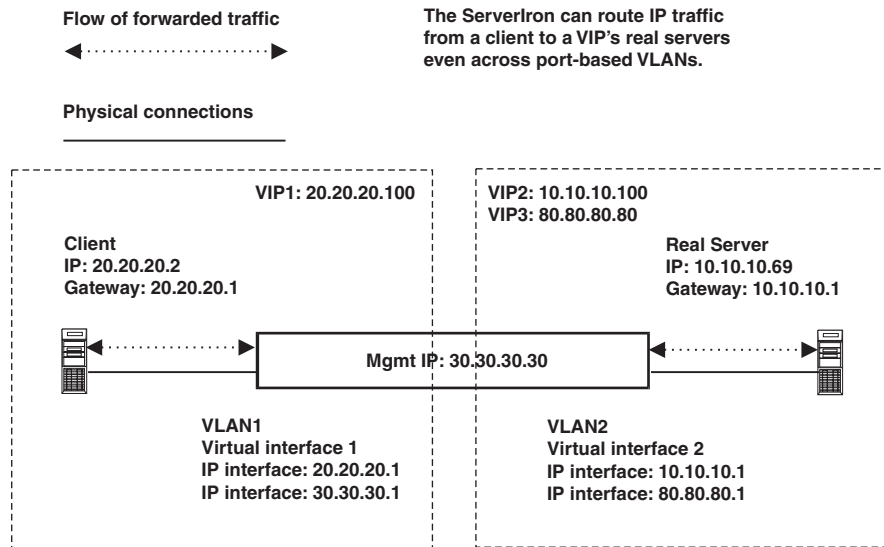
You also can use this type of IP forwarding configuration for Web Switching.

### Forwarding SLB or Web Switching Traffic Between Sub-nets in Different Broadcast Domains

Figure 13.7 on page 13-26 shows a configuration where an SLB client and the real server are in different sub-nets and also in separate port-based VLANs. Each port-based VLAN is a separate Layer 2 broadcast domain.

When you configure an IP interface for a VIP's sub-net, unless one of the VLANs has a host that is in the same sub-net as the VIP, you can configure the interface in any of the VLANs. If a VLAN has a host that is in the same sub-net as the VIP, add the IP interface for the VIP's sub-net to the VLAN that contains the host in the same sub-net.

For example, IP interface 80.80.80.1, which provides Layer 3 access for VIP 80.80.80.80, can be in VLAN 1 or VLAN 2. However, IP interface 20.20.20.1 must be in VLAN 1, since there is an IP host in the 20.20.20.x sub-net that is attached to VLAN 1.

**Figure 13.7 VIP and real servers in different sub-nets in different VLANs**


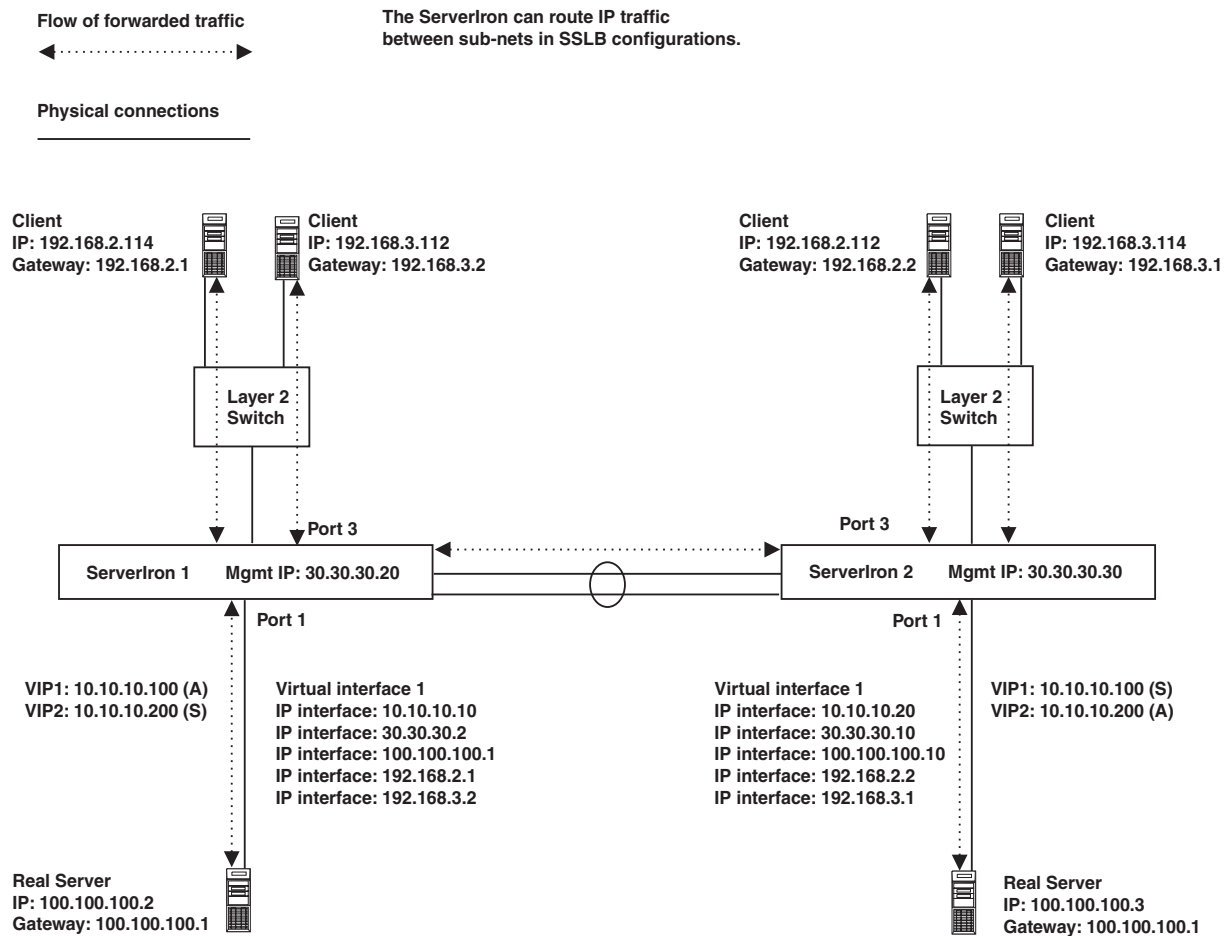
To configure IP forwarding on the ServerIron in Figure 13.7, enter the following commands.

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 30.30.30.30
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip address 30.30.30.1 255.255.255.0
ServerIron(config-vif-1)# ip address 20.20.20.1 255.255.255.0
ServerIron(config)# vlan 2
ServerIron(config-vlan-2)# untagged ethernet 13 to 24
ServerIron(config-vlan-2)# router-interface ve 2
ServerIron(config-vlan-2)# interface ve 2
ServerIron(config-vif-2)# ip address 10.10.10.1 255.255.255.0
ServerIron(config-vif-2)# ip address 80.80.80.1 255.255.255.0
ServerIron(config-vif-2)# exit
ServerIron(config)# ip forward
```

## Forwarding Symmetric SLB Traffic Between Sub-nets

Symmetric SLB (SSLB) allows multiple ServerIrons to actively load balance for specific VIPs while simultaneously serving as hot standbys for other VIPs. Each VIP is actively load balanced by just one of the ServerIrons at time. You can use IP forwarding in SSLB configurations.

Figure 13.8 on page 13-27 shows an SSLB configuration that uses IP forwarding. ServerIron 1 is the default active ServerIron for VIP1 but is the standby for VIP2. Likewise, ServerIron 2 is the default active ServerIron for VIP2 but is the standby for VIP1. The SSLB priority of a VIP determines which ServerIron is the default active ServerIron for that VIP.

**Figure 13.8 SSLB for ServerIrons and VIPs in different sub-nets – without routers**

Notice that each ServerIron is configured with an IP interface in each of the VIP's sub-nets. This is required for SSLB. The ServerIron must have an IP interface in each of the SSLB VIPs' sub-nets.

The link between the two ServerIrons in these examples is a trunk group. The link is not required to be a trunk group, but using a trunk group enhances the resiliency of the design by adding another layer of redundancy.

**NOTE:** This configuration requires source NAT, to ensure that reply traffic from a real server passes back through the correct ServerIron.

#### Commands for ServerIron 1

To configure IP forwarding on ServerIron 1 in Figure 13.8, enter the following commands.

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 30.30.30.20
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip address 10.10.10.10 255.255.255.0
ServerIron(config-vif-1)# ip address 30.30.30.2 255.255.255.0
ServerIron(config-vif-1)# ip address 100.100.100.1 255.255.255.0
ServerIron(config-vif-1)# ip address 192.168.2.1 255.255.255.0
ServerIron(config-vif-1)# ip address 192.168.3.2 255.255.255.0
```

```
ServerIron(config-vif-1)# exit
ServerIron(config)# ip forward
```

### **Commands for ServerIron 2**

To configure IP forwarding on ServerIron 2 in Figure 13.8, enter the following commands.

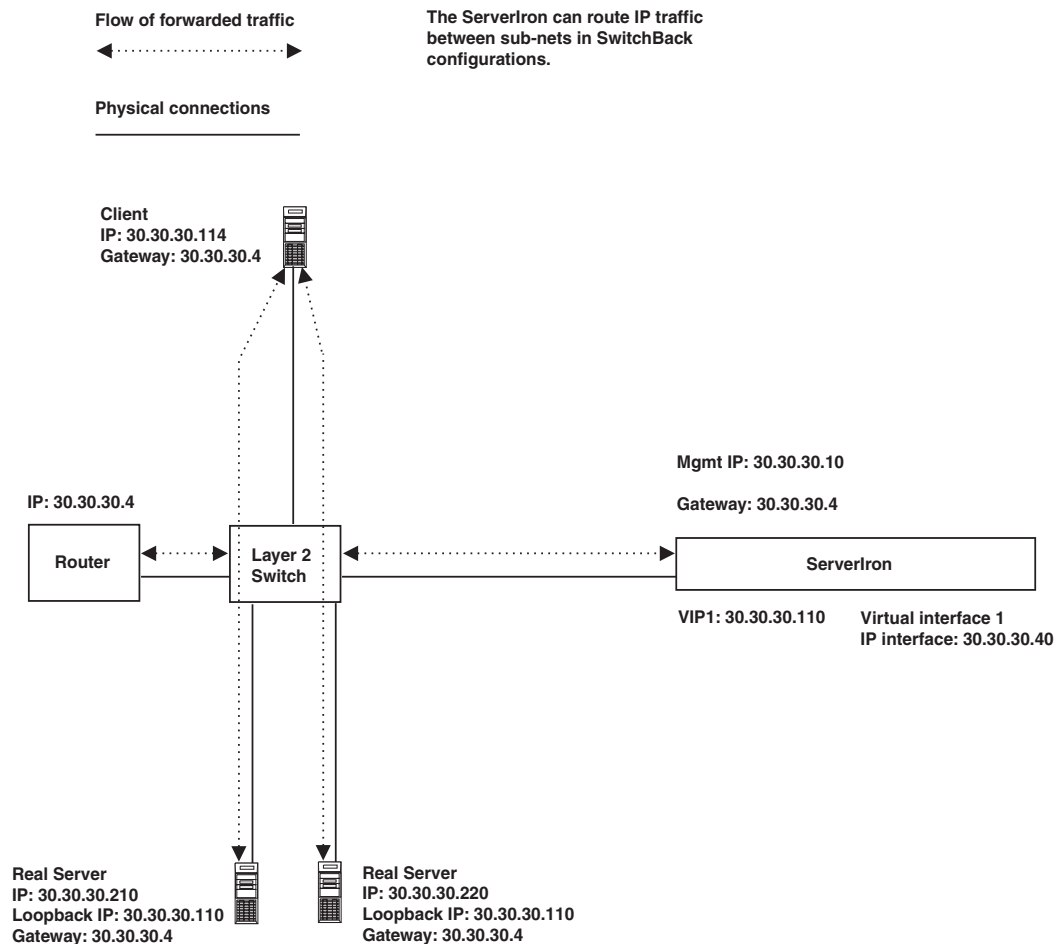
```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 30.30.30.30
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip address 10.10.10.20 255.255.255.0
ServerIron(config-vif-1)# ip address 30.30.30.10 255.255.255.0
ServerIron(config-vif-1)# ip address 100.100.100.10 255.255.255.0
ServerIron(config-vif-1)# ip address 192.168.2.2 255.255.255.0
ServerIron(config-vif-1)# ip address 192.168.3.1 255.255.255.0
ServerIron(config-vif-1)# exit
ServerIron(config)# ip forward
```

### **Forwarding SwitchBack Traffic Between Sub-nets**

You can configure a ServerIron to use IP forwarding in SwitchBack (direct server return) configurations. Figure 13.9 on page 13-29 shows a SwitchBack configuration in which the ServerIron and the clients are connected at Layer 2, through a Layer 2 Switch. The ServerIron has IP interfaces to place it in the following sub-nets:

- The client's sub-net
- The sub-net containing the ServerIron's default gateway
- The sub-net containing the real servers

Figure 13.9 SwitchBack configuration with ServerIron and client in different sub-nets



To configure the real servers for SwitchBack, configure a loopback interface on each real server and assign the VIP addresses to the loopback interface. The loopback interface enables the real server to respond to client requests directed at the VIPs, while at the same time keeping the real server hidden. The loopback interface responds to unicast traffic directed to it, but does not respond to ARP requests. The ServerIron responds to pings and ARPs for the VIPs. Thus, an attempt to obtain the real server's MAC address by ARPing a VIP does not succeed.

Here are the CLI commands for configuring IP forwarding on the ServerIron in Figure 13.9.

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 30.30.30.10
ServerIron(config)# ip default-gateway 30.30.30.4
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip address 30.30.30.40 255.255.255.0
ServerIron(config-vif-1)# exit
ServerIron(config)# ip forward
```

**NOTE:** The command **ip route 0.0.0.0 0.0.0.0 30.30.30.4** can be used in place of the **ip default-gateway 30.30.30.4**. Both commands configure the same system parameter.

## Forwarding in an SLB Hot-Standby Configuration

You can use IP forwarding in SLB hot-standby configurations. A hot-standby configuration adds fault tolerance to an SLB configuration by allowing the active and standby ServerIrons to share IP addresses. For example, you can provide fault tolerance for the default gateway addresses used by clients and real servers, by configuring the default gateway addresses as shared IP interfaces on the ServerIrons. If the active ServerIron becomes unavailable, the standby ServerIron takes over for the unavailable ServerIron and provides service for the shared IP interfaces. As a result, the clients and real servers can still reach their default gateways.

When you configure hot-standby, make the configurations of the two ServerIrons identical except for the management IP address (and optionally the hostname). On each of the ServerIrons, configure a standby IP interface for each of the following:

- The client's sub-net
- The real servers' sub-net(s)

---

**NOTE:** To simplify configuration, complete the configuration of one of the ServerIrons and save the changes to the startup-config file. Then copy the startup-config file to a TFTP server, edit the IP address, then copy it to the other ServerIron.

---

---

**NOTE:** The Spanning Tree Protocol (STP) interferes with the hot standby communication between the two ServerIrons and must be disabled. Although some backup topologies can appear to result in a logical loop, the backup ServerIron does not forward traffic. Therefore, no loop occurs.

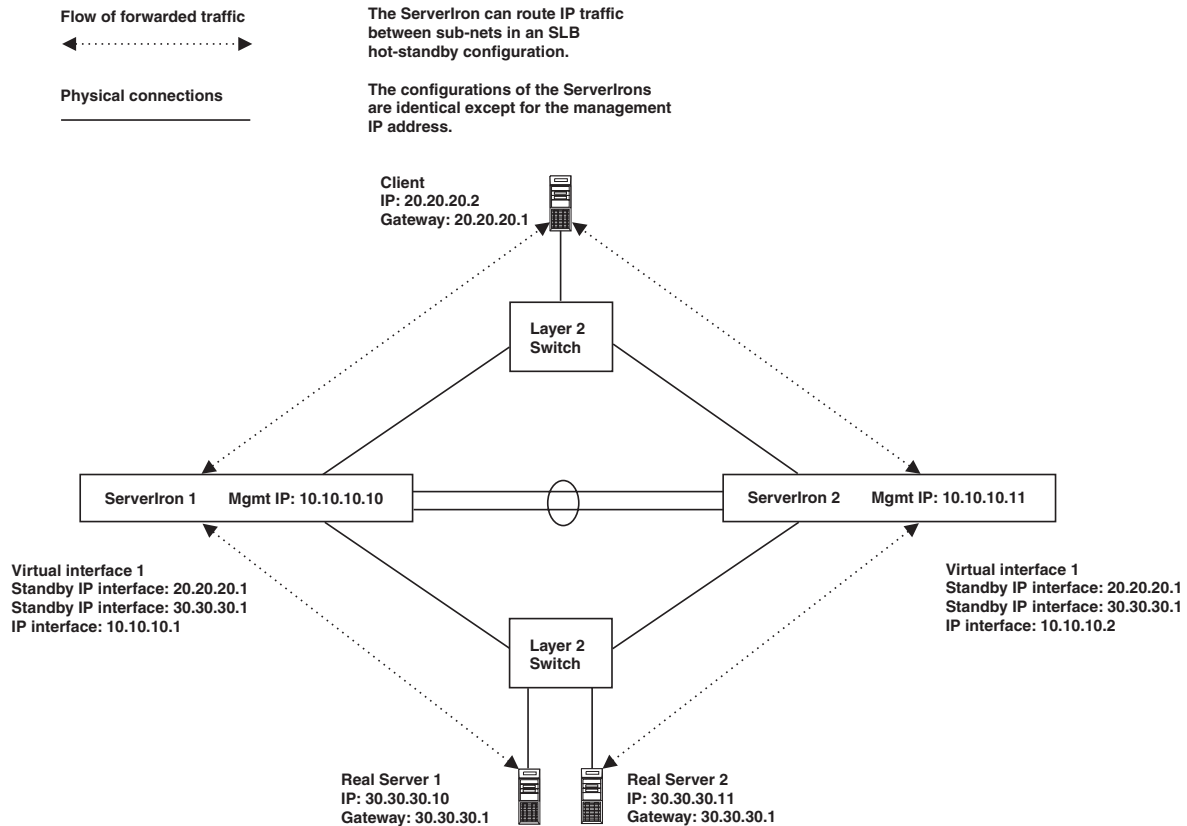
---

Figure 13.10 shows an example of a hot-standby configuration that uses IP forwarding. In this example, each ServerIron has two standby IP interfaces, which are the shared IP addresses. One of the standby interfaces is the client's default gateway. The other standby interface is the real servers' default gateway. Each ServerIron also has an IP interface that is not a standby interface. The ServerIrons do not share the IP interfaces that are not standby interfaces.

You also can configure IP interfaces that are not shared. In this case, the interface is available only on the ServerIron on which you configure the interface.



Figure 13.10 SLB hot-standby configuration



Here are the CLI commands for configuring the ServerIrons. The only difference between the two sets of commands is the IP management address. Notice that the **ip standby-address** command is used to add the IP interfaces, instead of the **ip address** command. The **ip standby-address** command allows the address to be shared by the two ServerIrons so that the standby ServerIron can provide service for the address if the active ServerIron becomes unavailable.

#### Commands for ServerIron 1

Here are the CLI commands for configuring IP forwarding on the ServerIron 1 in Figure 13.10.

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 10.10.10.10
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip standby-address 20.20.20.1 255.255.255.0
ServerIron(config-vif-1)# ip standby-address 30.30.30.1 255.255.255.0
ServerIron(config-vif-1)# ip address 10.10.10.1 255.255.255.0
ServerIron(config-vif-1)# exit
ServerIron(config)# ip forward
```

#### Commands for ServerIron 2

Here are the CLI commands for configuring IP forwarding on the ServerIron 2 in Figure 13.10.

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 10.10.10.11
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
```

```
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip standby-address 20.20.20.1 255.255.255.0
ServerIron(config-vif-1)# ip standby-address 30.30.30.1 255.255.255.0
ServerIron(config-vif-1)# ip address 10.10.10.2 255.255.255.0
ServerIron(config-vif-1)# exit
ServerIron(config)# ip forward
```

## Forwarding Traffic Between Sub-nets in a Transparent Cache Switching Configuration

You can use IP forwarding in Transparent Cache Switching (TCS) configurations. Figure 13.11 on page 13-32 shows an example of a TCS configuration in which a client, a file server, and a cache server are in different sub-nets. When TCS is enabled, the ServerIron redirects requests from the client for the file server, sending the requests to the cache server instead of the file server. Although this example shows a file server, TCS also can redirect client requests for the Internet.

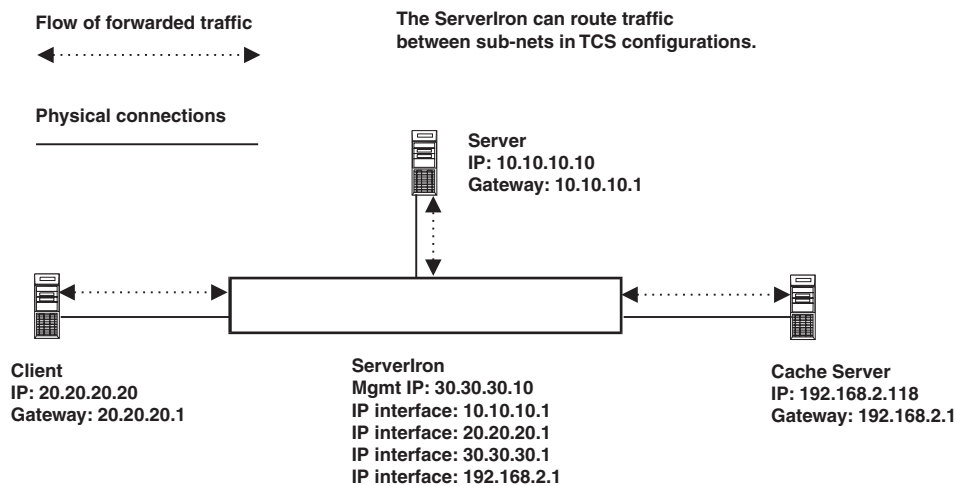
In this configuration, the ServerIron has four IP interfaces, one for the management IP address and one for the sub-nets the client, file server, and cache server are in.

---

**NOTE:** Without IP forwarding, this type of configuration can require the Cache Route Optimization (CRO) feature and the policy-based cache failover (CFO) feature. With IP forwarding, since the ServerIron is acting as the default gateway for all three devices, the CRO and CFO features are not required.

---

**Figure 13.11 TCS configuration with ServerIron and cache server in different sub-nets – without a router**



Here are the CLI commands for configuring IP forwarding on the ServerIron in Figure 13.11.

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 30.30.30.10
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip address 10.10.10.1 255.255.255.0
ServerIron(config-vif-1)# ip address 20.20.20.1 255.255.255.0
ServerIron(config-vif-1)# ip address 30.30.30.1 255.255.255.0
ServerIron(config-vif-1)# ip address 192.168.2.1 255.255.255.0
ServerIron(config-vif-1)# exit
ServerIron(config)# ip forward
```

## Forwarding Traffic in a Network Address Translation Configuration

Figure 13.12 on page 13-33 shows an example of a ServerIron configured for Network Address Translation (NAT). Generally, NAT is used to translate private addresses into public addresses so that clients in a private sub-net can communicate with devices that use public (Internet) addresses.

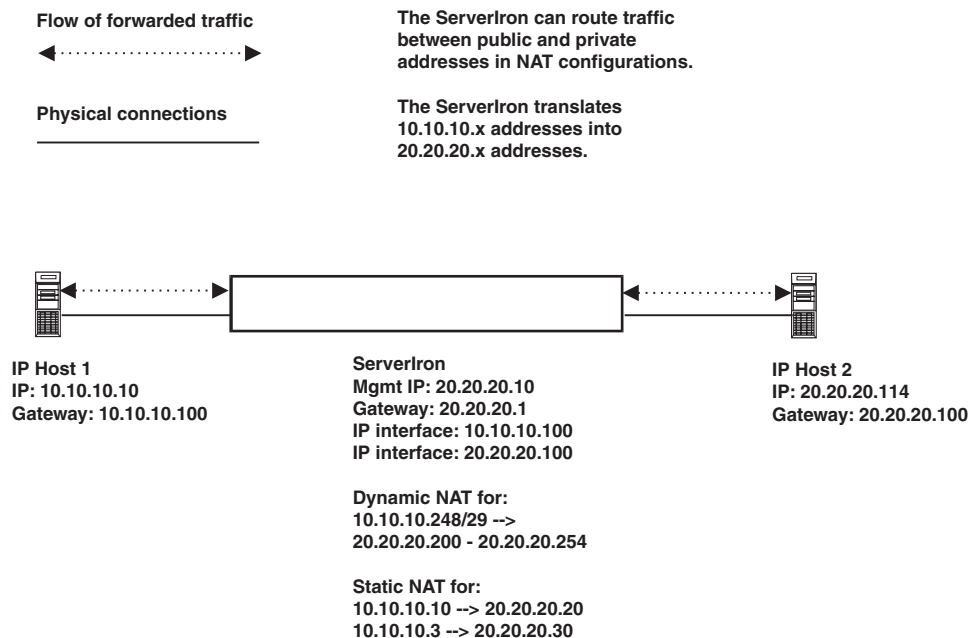
In this example, the ServerIron is configured to translate addresses in the 10.10.10.248/29 sub-net into addresses in the 20.20.20.x/24 sub-net. Most of the translations are accomplished dynamically using an address pool. However, static translations are used for two other addresses.

---

**NOTE:** You cannot use the same IP address as a NAT address and as an IP interface. For example, in this configuration, you cannot configure 20.20.20.100 as a NAT address.

---

**Figure 13.12 NAT configuration using IP forwarding**



Here are the CLI commands for configuring IP forwarding on the ServerIron in Figure 13.12. The **ip route** command adds a static IP route to provide a path to the ServerIron's default gateway. The next-hop address specified in the route is the IP address of the next-hop gateway.

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 20.20.20.10
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip address 10.10.10.100 255.255.255.0
ServerIron(config-vif-1)# ip address 20.20.20.100 255.255.255.0
ServerIron(config-vif-1)# exit
ServerIron(config)# ip route 0.0.0.0 0.0.0.0 20.20.20.1
ServerIron(config)# ip forward
```

## Forwarding Traffic in a GSLB Configuration

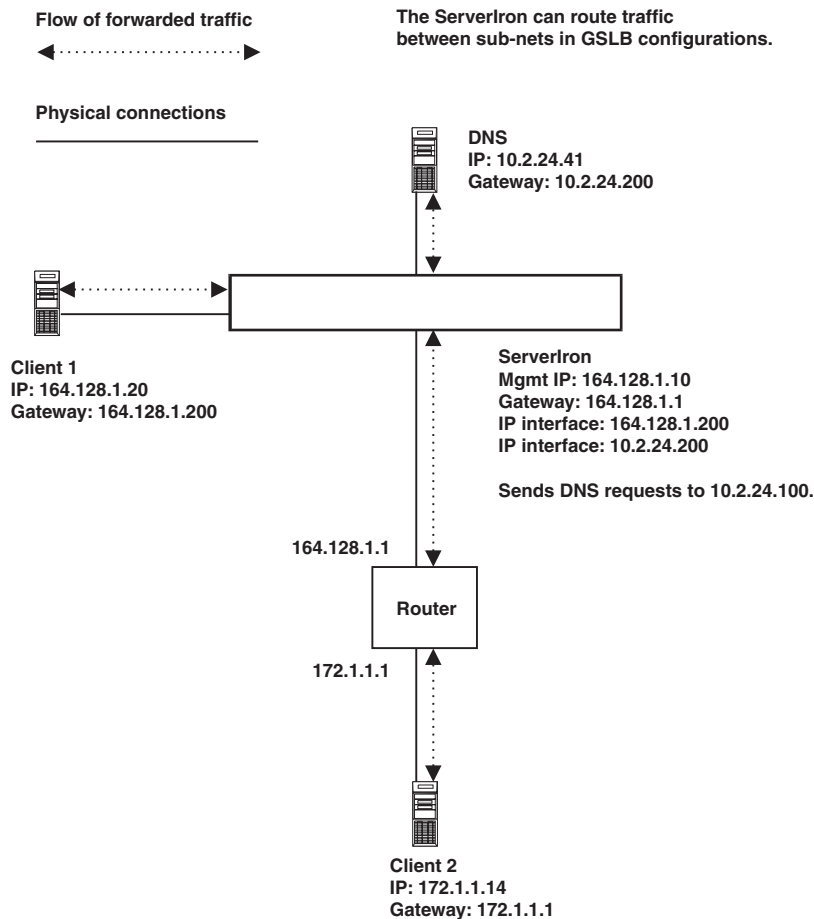
Global Server Load Balancing (GSLB) enables a ServerIron to add intelligence to authoritative Domain Name Servers (DNSs) by serving as a proxy to the servers. As a DNS proxy, the GSLB ServerIron evaluates the server IP addresses in the DNS replies from the DNS for which the ServerIron is a proxy. Based on the results of the

evaluation, the GSLB ServerIron can change the order of the addresses in the reply so that the “best” host address for the client is on top.

You also can configure a simplified variety of GSLB called Global IP. A Global IP configuration does not use an external DNS. Instead, the ServerIron provides the basic DNS service.

Figure 13.13 shows an example of a GSLB configuration that uses an external DNS, and uses IP forwarding on the ServerIron.

**Figure 13.13 GSLB configuration using IP forwarding**



Here are the CLI commands for configuring IP forwarding on the ServerIron in Figure 13.13. The **ip route** command adds a static IP route to provide a path to the ServerIron’s default gateway.

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 164.128.1.10
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip address 164.128.1.200 255.255.255.0
ServerIron(config-vif-1)# ip address 10.2.24.200 255.255.255.0
ServerIron(config-vif-1)# exit
ServerIron(config)# ip route 0.0.0.0 0.0.0.0 164.128.1.1
ServerIron(config)# ip forward
```

## Forwarding Traffic in a High-Availability FWLB Configuration

Figure 13.14 on page 13-36 shows an example of an IronClad (high-availability) FWLB configuration using IP forwarding. A typical IronClad FWLB configuration consists of two hot-standby pair of ServerIrons. One pair load balances traffic coming into the firewalls from external, unsecured hosts. The other pair load balances traffic coming from the secured (internal) hosts.

Although the configuration steps for an active-standby pair of ServerIrons differs between SLB hot-standby and IronClad FWLB, the IP forwarding configuration requirements are the same. In each case, you need to configure an identical set of standby IP interfaces on both the active and standby ServerIron.

Each ServerIron has two static IP routes. One of the routes is a default route and has the ServerIron's default gateway address as the next-hop address. The other route goes to the sub-net of the IP host that is on the same side of the firewalls as the ServerIron.

---

**NOTE:** Each ServerIron is configured to use one of the firewalls as its default gateway. Even though you can configure only one default gateway for a ServerIron, the ServerIron performs load balancing for through traffic. If a ServerIron's default gateway goes down (the firewall goes down), through traffic is still load balanced to the remaining firewall(s), and you can still access the management IP address on the ServerIron. In this case, the traffic takes an alternate path, supported by the always-active feature used in this example.

---

## Differences From Configurations Without IP Forwarding

Some of the configuration tasks that are required when you configure FWLB without IP forwarding are not required when you use IP forwarding:

- Static MAC entries – Configurations without IP forwarding require you to configure a static MAC entry for each firewall interface to which the ServerIron is connected. When you use IP forwarding, the static MAC entries are not required.
- Static ARP entries – Configurations without IP forwarding require you to configure a static ARP entry for the firewall on the internal and external routers, since the firewall is the next Layer 3 routing hop for traffic forwarded by one of these routers. When you use IP forwarding, the ServerIron itself is the next routing hop for traffic forwarded by one of these routers; thus, static ARP entries are not required.
- Port numbers in firewall paths – Without IP forwarding, you must indicate the ServerIron port connected to each path through the firewalls. If you move a cable, you must also reconfigure the path with the new port number. When you use IP forwarding, you can use the wildcard “255” instead of a specific port number when you configure a firewall, in which case you can move cables without needing to reconfigure the paths.

---

**NOTE:** This applies only to firewall paths. You must specify the port number when configuring a router path, regardless of whether you use IP forwarding.

---

## Configuration Details

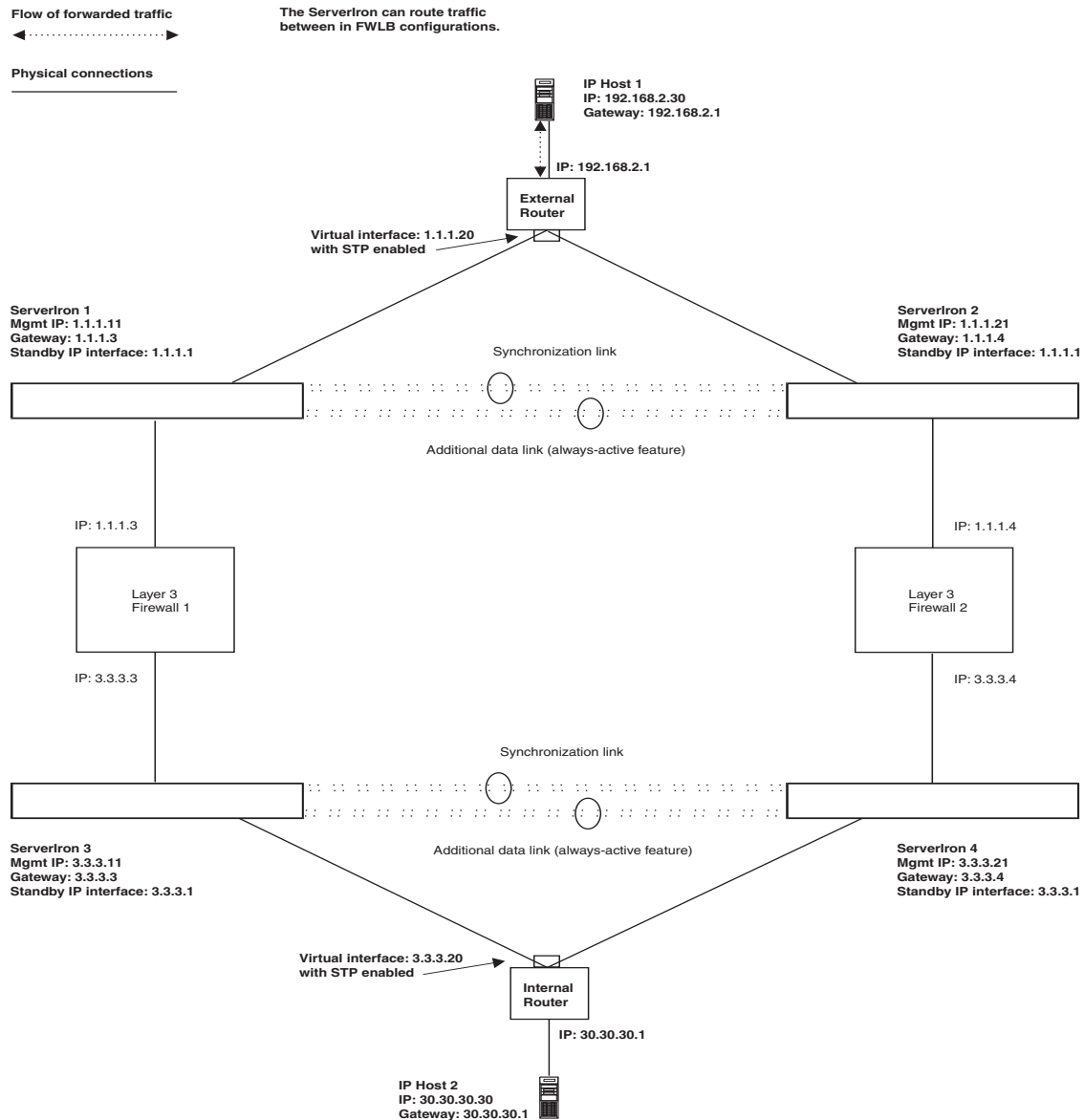
The Spanning Tree Protocol (STP) is enabled on the virtual interfaces of the internal and external routers.

This configuration in this example requires the following routes:

- External router
  - Route to 3.3.3.x/24 through 1.1.1.1
  - Route to 30.30.30.x/24 through 1.1.1.1
- Internal router
  - Route to 1.1.1.x/24 through 3.3.3.1
  - Route to 192.168.2.x/24 through 3.3.3.1
- Firewall 1 and Firewall 2
  - Route to 192.168.2.x/24 through 1.1.1.1

- Route to 30.30.30.x/24 through 3.3.3.1
- ServerIron 1 and ServerIron 2 (external ServerIrons)
  - Default route the directly connected firewall interface
  - Route to 192.168.2.x/24 through 1.1.1.20
- ServerIron 3 and ServerIron 4 (internal ServerIrons)
  - Default route the directly connected firewall interface
  - Route to 30.30.30.x/24 through 3.3.3.20

**Figure 13.14 High-availability FWLB configuration using IP forwarding**



Here are the CLI commands for configuring IP forwarding on the ServerIrons in Figure 13.14.

#### Commands for ServerIron 1

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
```

```
ServerIron(config)# ip address 1.1.1.11
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip standby-address 1.1.1.1 255.255.255.0
ServerIron(config-vif-1)# exit
ServerIron(config)# ip route 0.0.0.0 0.0.0.0 1.1.1.3
ServerIron(config)# ip route 192.168.2.0 255.255.255.0 1.1.1.20
ServerIron(config)# ip forward
```

#### **Commands for ServerIron 2**

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 1.1.1.21
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip standby-address 1.1.1.1 255.255.255.0
ServerIron(config-vif-1)# exit
ServerIron(config)# ip route 0.0.0.0 0.0.0.0 1.1.1.4
ServerIron(config)# ip route 192.168.2.0 255.255.255.0 1.1.1.20
ServerIron(config)# ip forward
```

#### **Commands for ServerIron 3**

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 3.3.3.11
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip standby-address 3.3.3.1 255.255.255.0
ServerIron(config-vif-1)# exit
ServerIron(config)# ip route 0.0.0.0 0.0.0.0 3.3.3.3
ServerIron(config)# ip route 30.30.30.0 255.255.255.0 3.3.3.20
ServerIron(config)# ip forward
```

#### **Commands for ServerIron 4**

```
ServerIron> enable <password, if required>
ServerIron# configure terminal
ServerIron(config)# ip address 3.3.3.21
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
ServerIron(config-vlan-1)# interface ve 1
ServerIron(config-vif-1)# ip standby-address 3.3.3.1 255.255.255.0
ServerIron(config-vif-1)# exit
ServerIron(config)# ip route 0.0.0.0 0.0.0.0 3.3.3.4
ServerIron(config)# ip route 30.30.30.0 255.255.255.0 3.3.3.20
ServerIron(config)# ip forward
```





---

## Chapter 14

# Configuring Network Address Translation

You can configure the ServerIron to perform standard **Network Address Translation (NAT)**. NAT enables private IP networks that use nonregistered IP addresses to connect to the Internet. Configure NAT on the Foundry device at the border of an inside network and an outside network (such as the Internet). NAT translates the internal local addresses to globally unique IP addresses before sending packets to the outside network. NAT also allows a more graceful renumbering strategy for organizations that are changing service providers or voluntarily renumbering into Classless Interdomain Routing (CIDR) blocks.

---

**NOTE:** The standard NAT support described in this section provides IP address translation for hosts attached to private networks on the ServerIron, and is separate from the virtual IP address features provided for Server Load Balancing (SLB). For example, standard NAT is not related to source IP addresses used for multinetting the ServerIron, performing health checks on remote servers, and so on.

---

**NOTE:** NAT does not support hot-standby (active-standby) configurations. If failover to a standby ServerIron occurs, the active NAT sessions on the ServerIron that has become unavailable are ended. The sessions are not continued by the standby ServerIron.

---

Use NAT to translate your private IP addresses into globally unique IP addresses when communicating outside of your network.

---

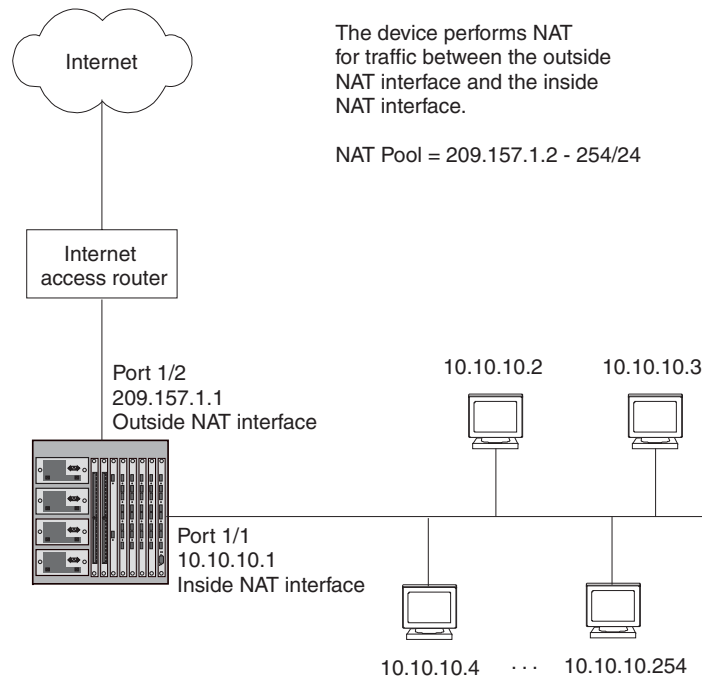
**NOTE:** You can configure up to 256 global IP addresses on a ServerIron.

---

A Foundry device configured for NAT must have an interface to the private network and an interface to a public network (for example, the Internet). In a typical environment, NAT is configured on the Foundry device between the private network and the Internet. When you configure a Foundry device for NAT, the device does not advertise the private networks to the Internet. However, the device can advertise route information received from the Internet to the private network.

Figure 14.1 shows an example of a network using NAT on a Foundry device. In this example, a ServerIron 800 is using NAT to translate traffic from the hosts on the 10.10.10.x/24 sub-net into public addresses from the address pool.

**Figure 14.1 Network Using Inside NAT**



The device performs NAT for traffic between the outside NAT interface and the inside NAT interface.

NAT Pool = 209.157.1.2 - 254/24

In this example, the ServerIron is configured to perform dynamic inside NAT for the private addresses in the 10.10.10.x/24 sub-net. The Foundry device uses the address pool 209.157.1.2/24 – 209.157.22.254/24 to map the private addresses to public addresses for traffic initiated by hosts in the 10.10.10.x/24 sub-net.

You can configure the following types of NAT:

- **Dynamic NAT** – Dynamic NAT maps private addresses to Internet addresses. The Internet addresses come from a pool of addresses that you configure. In the example in Figure 14.1, the pool is the range of addresses from 209.157.1.2/24 – 209.157.1.254/24. When you use dynamic NAT, the software uses a round robin technique to select a global IP address to map to a private address from a pool that you configure.
- **Static NAT** – Static NAT maps a particular global IP address (Internet IP address) with a particular private address. Use static NAT when you want to ensure that the software always maps the same public address to a given private address. For example, use static NAT when you want specific hosts in the private network to always use the same Internet address when communicating outside the private network.

**NOTE:** You can configure both dynamic and static NAT on the same device. When you configure both types of NAT, static NAT takes precedence over dynamic NAT. Thus, if you configure a static NAT translation for a private address, the ServerIron always uses that translation instead of creating a dynamic one.

## Port Address Translation

Normally, NAT maps each private address that needs to be routed to the outside network to a unique IP address from the pool. However, it is possible for the global address pool to have fewer addresses than the number of private addresses. In this case, you can configure the Foundry device to use Port Address Translation. **Port Address Translation** maps a client's IP address and TCP or UDP port number to both an IP address and a TCP or UDP port number. In this way, the Foundry device can map many private addresses to the same public address and use TCP or UDP port numbers to uniquely identify the private hosts.

**NOTE:** This type of feature is sometimes called Overloading an Inside Global Address.

In the example in Figure 14.1, the pool contains enough addresses to ensure that every host on the private network can be mapped to an Internet address in the pool. However, suppose the enterprise implementing this configuration has only 20 Internet addresses. For example, the pool might be 209.157.1.1/24 – 209.157.1.20/24. In this case, the pool does not contain enough addresses to ensure that all the hosts in the private network can be mapped to Internet addresses.

Without Port Address Translation, it is possible that the device will not be able to provide NAT for some hosts. However, with Port Address Translation, the device can provide NAT for all the hosts by using a unique TCP or UDP port number in addition to the IP address to map to each host. For example, the device can map the following addresses:

Inside address	Outside address
10.10.10.2:6000	209.157.1.2:1024
10.10.10.3:6000	209.157.1.2:1025
10.10.10.4:6000	209.157.1.2:1026

NAT is mapping the same global IP address to three different private addresses along with their TCP or UDP ports, but uses a different TCP or UDP port number for each private address to distinguish them. Notice that the Port Address Translation feature does not attempt to use the same TCP or UDP port number as in the client's packet.

The way NAT deals with the client's TCP or UDP port number depends on whether Port Address Translation is enabled:

- Port Address Translation enabled – NAT treats the client's IP address and TCP or UDP port number as a single entity, and uniquely maps that entity to another entity consisting of an IP address and TCP or UDP port number. The NAT entry the ServerIron creates in the NAT translation table therefore consists of an IP address plus a TCP or UDP port number. The ServerIron maintains the port type in the translation address:
  - If the client's packet contains a TCP port number, the ServerIron uses a TCP port in the translation address.
  - If the client's packet contains a UDP port, the ServerIron uses a UDP port in the translation address.

The ServerIron does not try to use the same TCP or UDP port number for the untranslated and translated addresses. Instead, the ServerIron maps the client IP address plus the TCP or UDP port number to a unique combination of IP address plus TCP or UDP port number. When the Foundry device receives reply traffic to one of these hosts, NAT can properly translate the Internet address back into the private address because the TCP or UDP port number in the translation address uniquely identifies the host.

To enable Port Address Translation, use the overload option when you configure the source list, which associates a private address range with a pool of Internet addresses. See "Configuring Dynamic NAT Parameters" on page 14-5.

- Port Address Translation disabled – The ServerIron translates only the client's IP address into another IP address and retains the TCP or UDP port number unchanged.

## Protocols Supported for NAT

NAT on the ServerIron supports the following protocols:

- ICMP
- UDP/TCP (generic)
- FTP
- VDOLive

- StreamWorks
- CU-SeeMe
- RealAudio and RealVideo
- RealMedia
- QuickTime
- Microsoft Media Services
- Web Theater (Vxtreme)

## Configuring NAT

To configure NAT, perform the following tasks:

- Configure the static address mappings, if needed. Static mappings explicitly map a specific private address to a specific Internet address to ensure that the addresses are always mapped together. Use static address mappings when you want to ensure that a specific host in the private network is always mapped to the Internet address you specify.
- Configure dynamic NAT parameters:
  - Configure a standard or extended ACL for each range of private addresses for which you want to provide NAT.
  - Configure a pool for each consecutive range of Internet addresses to which you want NAT to be able to map the private addresses specified in the ACLs. Each pool must contain a range with no gaps. If your Internet address space has gaps, configure separate pools for each consecutive range within the address space.
  - Associate a range of private addresses (specified in a standard or extended ACL) with a pool.
  - Optionally, enable the Port Address Translation feature. Use this feature if you have more private addresses that might need NAT than the Internet address pools contain.
- Enable NAT on the device.

---

**NOTE:** Named ACLs are not supported with NAT. You must use a numbered ACL.

---

In addition to the tasks listed above, you can modify the age timers for the address translation entries that ServerIron creates. See “Changing Translation Table Timeouts” on page 14-6 for information. For information about viewing the active NAT translations, see “Displaying the Active NAT Translations” on page 14-7.

The following sections provide procedures for configuring NAT. You can configure up to 256 unique global IP addresses on a ServerIron.

---

**NOTE:** The following sections show the CLI methods for configuring NAT. To use the Web management interface, see “Configuring NAT Using the Web Management Interface” on page 14-11.

---

### Configuring Static Address Translations

Use the following CLI method to configure static NAT.

#### *USING THE CLI*

To configure static NAT for an IP address, enter commands such as the following:

```
ServerIron(config)# ip nat inside source static 10.10.10.69 209.157.1.69
```

The commands in this example statically map the private address 10.10.10.69 to the Internet address 209.157.1.69.

**Syntax:** [no] ip nat inside source static <private-ip> <global-ip>

This command associates a specific private address with a specific Internet address. Use this command when you want to ensure that the specified addresses are always mapped together.

The **inside source** parameter specifies that the mapping applies to the private address sending traffic to the Internet.

The <private-ip> parameter specifies the private IP address.

The <global-ip> parameter specifies the Internet address. The ServerIron supports up to 255 global IP addresses.

Neither of the IP address parameters needs a network mask.

## Configuring Dynamic NAT Parameters

To configure dynamic NAT:

- Configure a standard or extended ACL for each private address range.

---

**NOTE:** Named ACLs are not supported with NAT. You must use a numbered ACL.

---

- Configure a pool for each consecutive range of Internet addresses.
- Associate private addresses (ACLs) with pools.
- Optionally, enable the Port Address Translation feature.

Use the following CLI method to configure dynamic NAT.

### USING THE CLI

To configure dynamic NAT, enter commands such as the following at the global CONFIG level of the CLI:

```
ServerIron(config)# access-list 1 permit 10.10.10.0/24
ServerIron(config)# ip nat pool OutAddrs 209.157.1.2 209.157.2.254 prefix-length 24
ServerIron(config)# ip nat inside source list 1 pool OutAddrs
```

These commands configure a standard ACL for the private sub-net 10.10.10.x/24, then enable inside NAT for the sub-net. Make sure you specify **permit** in the ACL, rather than **deny**. If you specify **deny**, the Foundry device will not provide NAT for the addresses.

**Syntax:** [no] ip nat pool <pool-name> <start-ip> <end-ip> netmask <ip-mask> | prefix-length <length>

This command configures the address pool.

The <pool-name> parameter specifies the pool name. The name can be up to 255 characters long and can contain special characters and internal blanks. If you use internal blanks, you must use quotation marks around the entire name.

The <start-ip> parameter specifies the IP address at the beginning of the pool range. Specify the lowest-numbered IP address in the range.

The <end-ip> parameter specifies the IP address at the end of the pool range. Specify the highest-numbered IP address in the range.

---

**NOTE:** The address range cannot contain any gaps. Make sure you own all the IP addresses in the range. If the range contains gaps, you must create separate pools containing only the addresses you own.

---

The **netmask** <ip-mask> | **prefix-length** <length> parameter specifies a classical sub-net mask (example: **netmask** 255.255.255.0) or the length of a Classless Interdomain Routing prefix (example: **prefix-length** 24). The ServerIron supports up to 255 global IP addresses.

**Syntax:** [no] ip nat inside source list <acl-id> pool <pool-name> [overload]

This command associates a private address range with a pool of Internet addresses and optionally enables the Port Address Translation feature.

The **inside source** parameter specifies that the translation applies to private addresses sending traffic to the Internet (**inside source**).

The **list** <acl-id> parameter specifies a standard or extended ACL. You can specify a numbered or named ACL.

---

**NOTE:** Named ACLs are not supported with NAT. You must use a numbered ACL.

---



---

**NOTE:** For complete standard and extended ACL syntax, see the “Using Access Control Lists (ACLs)” chapter of the *Foundry Switch and Router Installation and Basic Configuration Guide*.

---

The **pool** <pool-name> parameter specifies the pool. You must create the pool before you can use it with this command.

The **overload** parameter enables the Port Address Translation feature. Use this parameter if the IP address pool does not contain enough addresses to ensure NAT for each private address. The Port Address Translation feature conserves Internet addresses by mapping the same Internet address to more than one private address and using a TCP or UDP port number to distinguish among the private hosts. The ServerIron supports up to 50 IP addresses with this feature enabled.

## Enabling NAT

To enable NAT, perform the following steps:

- Configure global policies to allow the ServerIron to examine the Layer 4 TCP and UDP information in IP packets.
- Enable the NAT feature.

You must perform both steps for all NAT configurations. NAT parameters do not take effect until you perform these steps to enable NAT.

---

**NOTE:** On the ServerIron, you enable NAT globally. You cannot enable NAT on an individual interface basis.

---

To enable NAT, use the following CLI method.

### USING THE CLI

To enable NAT on the ServerIron, enter the following command at the global CONFIG level of the CLI:

```
ServerIron(config)# ip policy 1 cache tcp 0 global
ServerIron(config)# ip policy 2 cache udp 0 global
ServerIron(config)# ip nat inside
```

**Syntax:** ip policy <policy-num> cache tcp | udp 0 global

The <policy-num> value identifies the policy and can be a number from 1 – 64.

Each policy affects TCP or UDP traffic, so you must specify **tcp** or **udp**.

The value 0 following the **tcp | udp** parameter specifies that the policy applies to all ports of the specified type (TCP or UDP). In this command, “0” is equivalent to “any port number”. For NAT, you must specify “0”.

**Syntax:** [no] ip nat inside

This command enables inside NAT.

## Changing Translation Table Timeouts

The NAT translation table contains all the currently active NAT translation entries on the device. An active entry is one that the ServerIron created for a private address when that client at that address sent traffic to the Internet. NAT performs the following steps to provide an address translation for a source IP address:

- The feature looks in the NAT translation table for an active NAT entry for the translation. If the table contains an active entry for the session, the ServerIron uses that entry.

- If NAT does not find an active entry in the NAT translation table, NAT creates an entry and places the entry in the table. The entry remains in the table until the entry times out.

Each NAT entry remains in the NAT translation table until the entry ages out.

- **Dynamic timeout** – This age timer applies to all entries (static and dynamic) that do not use Port Address Translation. The default is 120 seconds.
- **UDP timeout** – This age timer applies to entries that use Port Address Translation based on UDP port numbers. The default is 120 seconds.
- **TCP timeout** – This age timer applies to entries that use Port Address Translation based on TCP port numbers. The default is 120 seconds.

---

**NOTE:** This timer applies only to TCP sessions that do not end “gracefully”, with a TCP FIN or TCP RST.

---

- **TCP FIN/RST timeout** – This age timer applies to TCP FIN (finish) and RST (reset) packets, which normally terminate TCP connections. The default is 120 seconds.

---

**NOTE:** This timer is not related to the TCP timeout. The TCP timeout applies to packets to or from a host address that is mapped to an global IP address and a TCP port number (Port Address Translation feature). The TCP FIN/RST timeout applies to packets that terminate a TCP session, regardless of the host address or whether Port Address Translation is used.

---

- **DNS timeout** – This age timer applies to connections to a Domain Name Server (DNS). The default is 120 seconds.

To change the timeout for a dynamic entry type, use the following CLI method.

#### *USING THE CLI*

To change the age timeout for all entries that do not use Port Address Translation to 1800 seconds (one half hour), enter a command such as the following at the global CONFIG level of the CLI:

```
ServerIron(config)# ip nat timeout 1800
```

**Syntax:** [no] ip nat translation timeout | udp-timeout | tcp-timeout | finrst-timeout | dns-timeout <secs>

Use one of the following parameters to specify the dynamic entry type:

- **timeout** – All entries that do not use Port Address Translation. The default is 120 seconds.
- **udp-timeout** – Dynamic entries that use Port Address Translation based on UDP port numbers. The default is 120 seconds.
- **tcp-timeout** – Dynamic entries that use Port Address Translation based on TCP port numbers. The default is 120 seconds.
- **finrst-timeout** – TCP FIN (finish) and RST (reset) packets, which normally terminate TCP connections. The default is 120 seconds.
- **dns-timeout** – Connections to a Domain Name Server (DNS). The default is 120 seconds.

The <secs> parameter specifies the number of seconds. For each entry type, you can enter a value from 1 – 3600.

## Displaying the Active NAT Translations

To display the currently active NAT translations, display the NAT translation table using the following CLI method.

---

**NOTE:** For information about the aging timer for NAT translation entries, see “Changing Translation Table Timeouts” on page 14-6.

---

## USING THE CLI

To display the currently active NAT translations, enter the following command at any level of the CLI:

```
ServerIron(config)# show ip nat translation
Pro Inside global      Inside local      Outside local      Outside global
--- 209.157.1.69       10.10.10.69       207.195.2.12       207.195.2.12
--- 209.157.1.72       10.10.10.2        207.195.4.69       207.195.4.69
```

**Syntax:** show ip nat translation

**NOTE:** On the ServerIron 400 and ServerIron 800, you can enter this command only when logged in to a WSM CPU. The command is not supported on the Main Processor CPU. To log in to a WSM CPU, see “Logging In to a WSM CPU” on page 4-9.

The **show ip nat translation** command shows the following information.

**Table 1: CLI Display of Active NAT Translations**

This Field...	Displays...
Pro	When Port Address Translation is enabled, this field indicates the protocol NAT is using to uniquely identify the host. NAT can map the same IP address to multiple hosts and use the protocol port to distinguish among the hosts. This field can have one of the following values: <ul style="list-style-type: none"> <li>tcp – In addition to this IP address, NAT is associating a TCP port with the host on the private network.</li> <li>udp – In addition to this IP address, NAT is associating a UDP port with the host on the private network.</li> </ul>
Inside global	The Internet address mapped to the private address listed in the Inside local field for inside NAT.
Inside local	The private address mapped to the Internet private address listed in the Inside global field for inside NAT.
Outside global	The destination of the traffic. If Port Address Translation is enabled, the TCP or UDP port also is shown. <b>Note:</b> Outside NAT is not supported in the current software release.
Outside local	The destination of the traffic. If Port Address Translation is enabled, the TCP or UDP port also is shown. <b>Note:</b> Outside NAT is not supported in the current software release.

## Displaying NAT Statistics

To display NAT statistics, use the following CLI method.

### USING THE CLI

To display the NAT statistics, enter the following command at any level of the CLI:

```
ServerIron(config)# show ip nat statistics
Total translations: 2 (1 static, 1 dynamic)
```



```

Hits: 2   Misses: 2
Expired translations: 4
Dynamic mappings:
  pool OutAdds: netmask 255.255.255.0
    start 209.157.1.2 end 209.157.1.254
    total addresses 252

```

**Syntax:** show ip nat statistics

**NOTE:** On the ServerIron 400 and ServerIron 800, you can enter this command only when logged in to a WSM CPU. The command is not supported on the Main Processor CPU. To log in to a WSM CPU, see “Logging In to a WSM CPU” on page 4-9.

The **show ip nat statistics** command shows the following information.

**Table 2: CLI Display of NAT Statistics**

This Field...	Displays...
Total translations	The number of translations that are currently active. This number changes when translations are added or age out. To display the currently active translations, enter the <b>show ip nat translation</b> command.
Hits	The number of times NAT searched the translation table for a NAT entry and found the needed entry. (To optimize performance, NAT looks in the NAT table for an existing entry for a given translation before creating an entry for that translation.)
Misses	The number of times NAT did not find a needed entry in the translation table. When this occurs, NAT creates the needed entry and places it in the table.
Expired translations	The total number of dynamic translations that have aged of the translation table since the Foundry device was booted.
Dynamic mappings	<p>Lists the dynamic translation parameters configured for the device. The following information is displayed:</p> <ul style="list-style-type: none"> <li>pool – The name of the pool from which the address used for the translation was drawn.</li> <li>mask – The sub-net mask or prefix used for addressed in the pool.</li> <li>start – The beginning (lowest) IP address in the pool.</li> <li>end – The ending (highest) IP address in the pool.</li> <li>total addresses – The total number of active address translations that are based on addresses in this pool.</li> </ul> <p>In addition, if the pool uses the Port Address Translation feature, the word “overloaded” appears at the end of this row.</p>

## Clearing Translation Table Entries

In addition to the aging mechanism, the software allows you to manually clear entries from the NAT table. The software provides the following clear options:

- Clear all entries (static and dynamic)

- Clear an entry for a specific NAT entry based on the private and global IP addresses
- Clear an entry for a specific NAT entry based on the IP addresses and the TCP or UDP port number. Use this option when you are trying to clear specific entries created using the Port Address Translation feature.

To clear entries, use the following CLI method.

#### *USING THE CLI*

To clear all dynamic entries from the NAT translation table, enter the following command at the Privileged EXEC level of the CLI:

```
ServerIron# clear ip nat all
```

**Syntax:** clear ip nat all

To clear only the entries for a specific address entry, enter a command such as the following:

```
ServerIron# clear ip nat inside 209.157.1.43 10.10.10.5
```

This command clears the inside NAT entry that maps private address 10.10.10.5 to Internet address 209.157.1.43. Here is the syntax for this form of the command.

**Syntax:** clear ip nat inside <global-ip> <private-ip>

If you use Port Address Translation, you can selectively clear entries based on the TCP or UDP port number assigned to an entry by the feature. For example, the following command clears one of the entries associated with Internet address 209.157.1.44 but does not clear other entries associated with the same address.

```
ServerIron# clear ip nat inside 209.157.1.43 1081 10.10.10.5 80
```

The command above clears all inside NAT entries that match the specified global IP address, private IP address, and TCP or UDP ports.

**Syntax:** clear ip nat <protocol> inside <global-ip> <internet-tcp/udp-port> <private-ip> <private-tcp/udp-port>

The <protocol> parameter specifies the protocol type and can be **tcp** or **udp**.

---

**NOTE:** These commands are not supported on the ServerIron 400 or ServerIron 800.

---

## NAT Debug Commands

To configure the ServerIron to display diagnostic information for NAT, enter a **debug ip nat** command.

**Syntax:** debug ip nat icmp | tcp | udp <ip-addr>

**Syntax:** debug ip nat transdata

The <ip-addr> parameter specifies an IP address. The address applies to packets with the address as the source or the destination. Specify 0.0.0.0 to enable the diagnostic mode for all addresses.

---

**NOTE:** The **debug ip nat** commands are not supported on the ServerIron 400 or ServerIron 800.

---

The following examples show sample output from **debug ip nat** commands. The first three examples show the output from the diagnostic mode for ICMP NAT, TCP NAT, and UDP NAT. The fourth command shows the output for the diagnostic mode for NAT translation requests.

```
ServerIron# debug ip nat icmp 0.0.0.0
NAT: icmp src 10.10.100.18 => trans 192.168.2.79 dst 204.71.202.127
NAT: 192.168.2.79 204.71.202.127 ID 35768 len 60 txfid 13 icmp (8/0/512/519)
NAT: 204.71.202.127 10.10.100.18 ID 11554 len 60 txfid 15 icmp (0/0/512/519)

ServerIron# debug ip nat tcp 0.0.0.0
NAT: tcp src 10.10.100.18:1473 => trans 192.168.2.78:8016 dst 192.168.2.158:53
NAT: 192.168.2.78:8016 192.168.2.158:53 flags S ID 57970 len 44 txfid 13
NAT: 192.168.2.158:53 10.10.100.18:1473 flags S A ID 22762 len 44 txfid 15
NAT: 192.168.2.78:8016 192.168.2.158:53 flags A ID 58226 len 40 txfid 13
NAT: 192.168.2.78:8016 192.168.2.158:53 flags A ID 58482 len 77 txfid 13
NAT: 192.168.2.158:53 10.10.100.18:1473 flags A ID 23018 len 42 txfid 15
NAT: 192.168.2.78:8016 192.168.2.158:53 flags A ID 58738 len 40 txfid 13
NAT: 192.168.2.158:53 10.10.100.18:1473 flags A ID 23274 len 131 txfid 15
NAT: 192.168.2.78:8016 192.168.2.158:53 flags FA ID 58994 len 40 txfid 13
NAT: 192.168.2.158:53 10.10.100.18:1473 flags A ID 23530 len 40 txfid 15
NAT: 192.168.2.158:53 10.10.100.18:1473 flags FA ID 23786 len 40 txfid 15
NAT: 192.168.2.78:8016 192.168.2.158:53 flags A ID 59250 len 40 txfid 13

ServerIron# debug ip nat udp 0.0.0.0
NAT: udp src 10.10.100.18:1561 => trans 192.168.2.79:65286 dst 192.168.3.11:53
NAT: 192.168.2.79:65286 192.168.3.11:53 ID 35512 len 58 txfid 13
NAT: 192.168.3.11:53 10.10.100.18:1560 ID 8453 len 346 txfid 15

ServerIron# debug ip nat transdata
NAT: icmp src 10.10.100.18:2048 => trans 192.168.2.79 dst 204.71.202.127
NAT: udp src 10.10.100.18:1561 => trans 192.168.2.79:65286 dst 192.168.3.11:53
NAT: tcp src 10.10.100.18:1473 => trans 192.168.2.78:8016 dst 192.168.2.158:53
```

To disable the NAT diagnostic mode, enter a command such as the following:

**Syntax:** undebug ip nat icmp | tcp | udp | transdata

```
ServerIron# undebug ip nat tcp
```

This command disables the diagnostic mode for NAT performed on TCP packets. NAT diagnostics for other types of packets remain enabled.

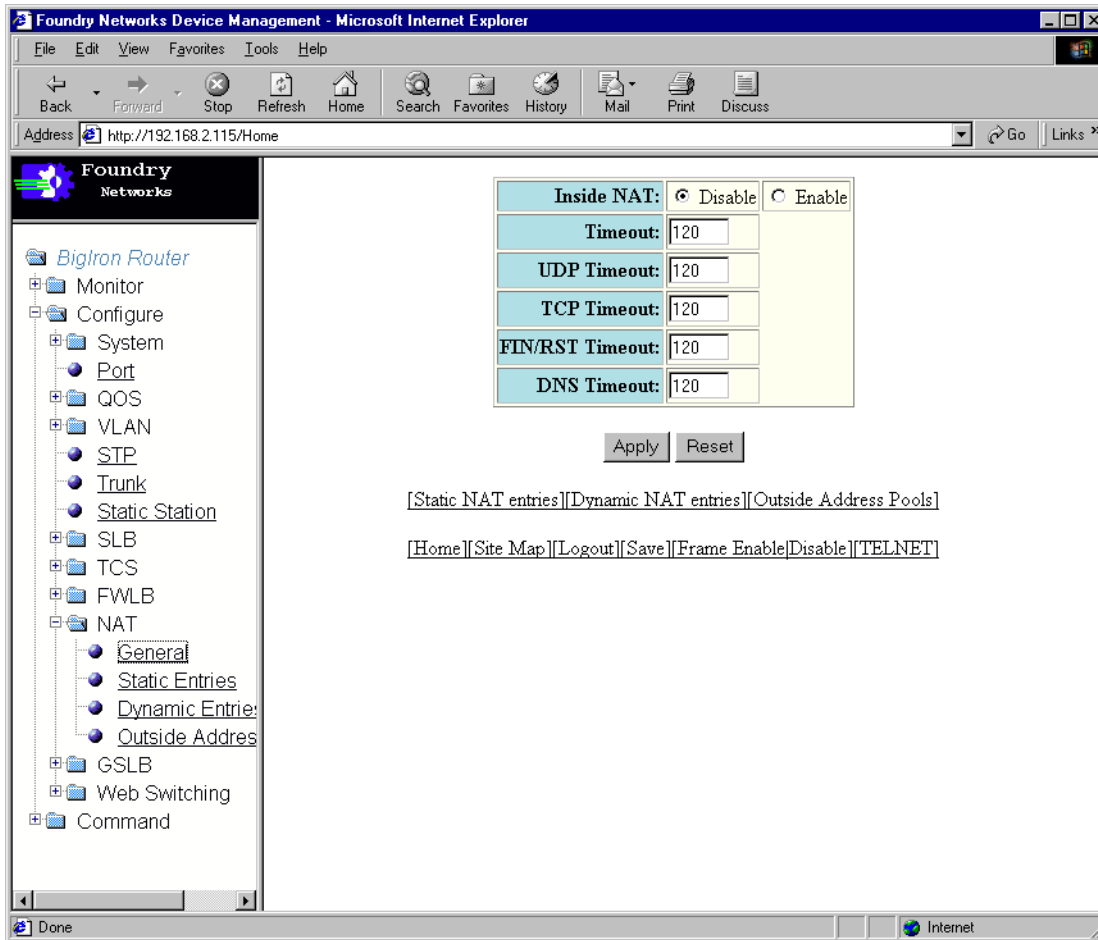
## Configuring NAT Using the Web Management Interface

To configure NAT using the Web management interface, use the following procedures.

### Configuring Global NAT Parameters

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to NAT in the tree view to expand the list of NAT option links.

4. Select the General link from the menu. The following panel will appear.



5. Click Enable next to Inside NAT to enable the feature.
6. Optionally, change NAT entry timers by editing the value in the corresponding field. Each NAT entry remains in the NAT translation table until the entry ages out. You can edit the following timers. For each timer, you can specify a value from 1 – 3600. The default for each timer is 120 seconds.
  - Timeout – This age timer applies to all entries (static and dynamic) that do not use Port Address Translation.
  - UDP timeout – This age timer applies to entries that use Port Address Translation based on UDP port numbers.
  - TCP timeout – This age timer applies to entries that use Port Address Translation based on TCP port numbers.

**NOTE:** This timer applies only to TCP sessions that do not end “gracefully”, with a TCP FIN or TCP RST.

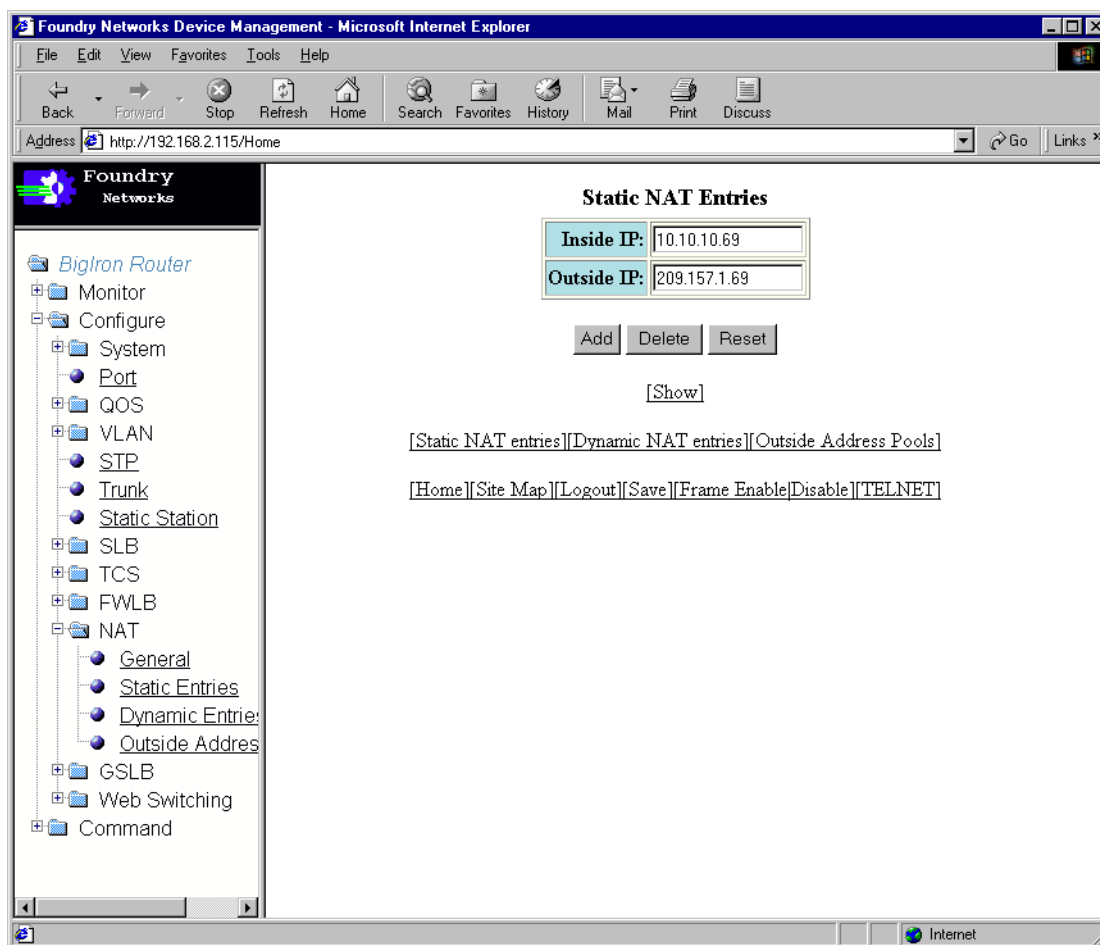
- FIN/RST timeout – This age timer applies to TCP FIN (finish) and RST (reset) packets, which normally terminate TCP connections.

**NOTE:** This timer is not related to the TCP timeout. The TCP timeout applies to packets to or from a host address that is mapped to an global IP address and a TCP port number (Port Address Translation feature). The TCP FIN/RST timeout applies to packets that terminate a TCP session, regardless of the host address or whether Port Address Translation is used.

- DNS timeout – This age timer applies to connections to a Domain Name Server (DNS).
7. Select the Apply button to apply the changes to the device's running-config.
  8. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Configuring Static NAT Entries

1. Log on to the device using a valid user name and password for read-write access.
2. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
3. Click on the plus sign next to NAT in the tree view to expand the list of NAT option links.
4. Select the Static Entries link from the menu. The following panel will appear.



5. Enter the inside IP address in the Inside IP field. This is a private IP address on your network.
6. Enter the outside IP address in the Outside IP field. This is the public (Internet) address that NAT translates the private address into before sending the private address' traffic to the Internet.

---

**NOTE:** Neither of the IP address parameters uses a network mask.

---

7. Select the Add button to add the addresses to the device's running-config.
8. Select the [Save](#) link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

## Configuring Dynamic NAT Entries

To configure dynamic NAT:

- Configure an Access Control List (ACL) for each private address range.

---

**NOTE:** Named ACLs are not supported with NAT. You must use a numbered ACL.

---

- Configure a pool for each consecutive range of Internet addresses.
- Associate the ACL with the pool and optionally enable the Port Address Translation feature.

## Configuring an ACL

---

**NOTE:** This procedure describes how to configure a standard ACL, which contains IP address and TCP or UDP source addresses. You also can use an extended ACL, which contains source and destination addresses. The dynamic NAT feature requires only the source information and works equally well with a standard or extended ACL.

---

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Configure in the tree view to display the list of configuration options.
3. Click on the plus sign next to System or IP to display more configuration options. You can access the ACL configuration panels from either location.
4. Select the [Standard ACL](#) link.
  - If the device does not already have some standard ACLs, the Standard ACL configuration panel is displayed, as shown in the following example.
  - Otherwise, if the device already has some standard ACLs, the Standard ACL table is displayed. This table lists the configured ACLs. Select the [Add Standard ACL](#) link to display the Standard ACL configuration panel, as shown in the following example.

**Standard ACL**

<b>Standard ACL Number:</b>	<input type="text" value="1"/>
<b>Action:</b>	<input checked="" type="radio"/> Permit <input type="radio"/> Deny
<b>IP Address:</b>	<input type="text" value="0.0.0.0"/>
<b>Subnet Mask:</b>	<input type="text" value="0.0.0.0"/>
<b>Host Name:</b>	<input type="text"/>
<b>Log:</b>	<input type="checkbox"/>

[\[Show\]](#)

[\[Home\]](#)
[\[Site Map\]](#)
[\[Logout\]](#)
[\[Save\]](#)
[\[Frame Enable\]](#)
[\[Disable\]](#)
[\[TELNET\]](#)

5. Change the ACL number in the Standard ACL Number field or use the ACL number displayed in the field.

---

**NOTE:** You cannot specify a name.

---

6. Select the ACL action. You can select Permit or Deny:

- Permit – Forwards traffic or allows management access for the specified IP source.
  - Deny – Drops traffic or denies management access for the specified IP source.
- 

**NOTE:** If the ACL is a forwarding ACL, the action forwards or drops the traffic. If the ACL is a management access ACL, the action permits or denies management access.

---

7. Enter the source information. You can enter the source IP address and network mask or the host name.
- If you enter the address, you also must enter the network mask. To specify “any”, enter “0.0.0.0”.
  - If you enter a host name instead of an IP address, when you click Add to add the ACL, the Web management interface sends a DNS query for the address. For the query to be successful, the device must have network access to a DNS server and the server must have an Address record for the host. In addition, the device must be configured with a DNS domain name and the IP address of the DNS server.
8. If you specified the Deny action, optionally enable logging by selecting the Log checkbox. If you enable logging for this ACL entry, the software generates Syslog entries for traffic that the ACL denies.
9. Click the Add button to save the ACL to the device’s running-config file.
10. Select the [IP Access Group](#) link.
- If the device does not already have some ACLs applied to interfaces, the IP Access Group configuration panel is displayed, as shown in the following example.
  - Otherwise, if the device already has some ACLs applied to interfaces, the IP Access Group table is displayed. Select the [Add](#) link to display the IP Access Group configuration panel, as shown in the following example.

**IP Access Group**

Slot:	1	Port:	1
Direction:	<input type="checkbox"/> In Bound <input type="checkbox"/> Out Bound		
ACL Number:	0		

[Add](#) [Delete](#) [Reset](#)

[\[Show\]](#)

[\[Home\]](#) [\[Site Map\]](#) [\[Logout\]](#) [\[Save\]](#) [\[Frame Enable\]](#) [\[Disable\]](#) [\[TELNET\]](#)

11. Select the Slot (if you are configuring a Chassis device) and port from the Slot and Port pulldown menus.
12. Specify the traffic direction to which the ACL applies. You can select one or both of the following:
- In Bound – The ACL applies to traffic received on the port from other devices.
  - Out Bound – The ACL applies to traffic this Foundry device queues for transmission on the port.
13. Enter the ACL number in the ACL Number field.
- 

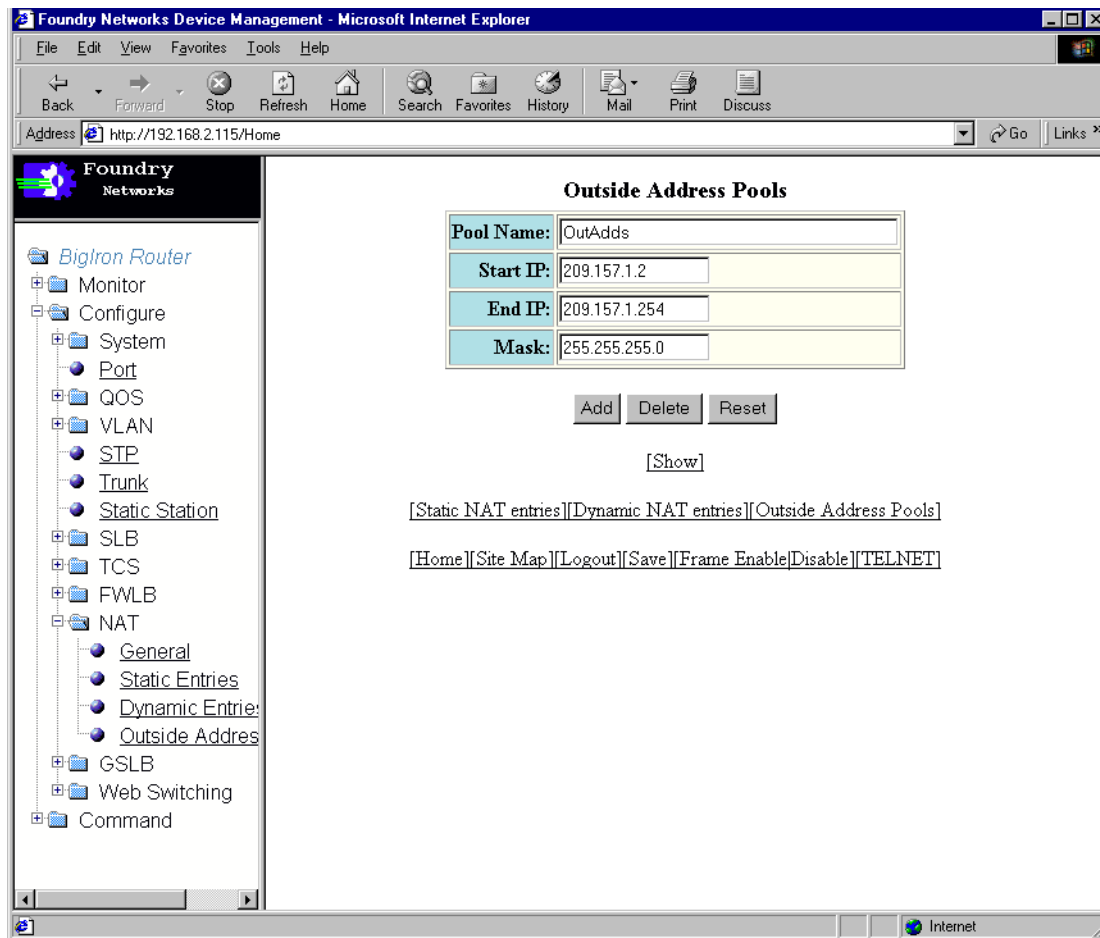
**NOTE:** You cannot specify a named ACL.

---

14. Click the Add button to save the ACL to the device’s running-config file.
15. Select the [Save](#) link at the bottom of the dialog. Select Yes when prompted to save the configuration change to the startup-config file on the device’s flash memory.
-

## Configuring a Pool

1. Select the Outside Address Pool link from the menu. The following panel will appear.



2. Enter the pool name in the Pool Name field.
3. Enter the starting IP address in the Start IP field. This is the IP address at the beginning of the range of addresses in the pool.
4. Enter the ending IP address in the End IP field. This is the IP address at the end of the range of addresses in the pool.

---

**NOTE:** The range cannot have gaps. All addresses within the specified range are members of the pool.

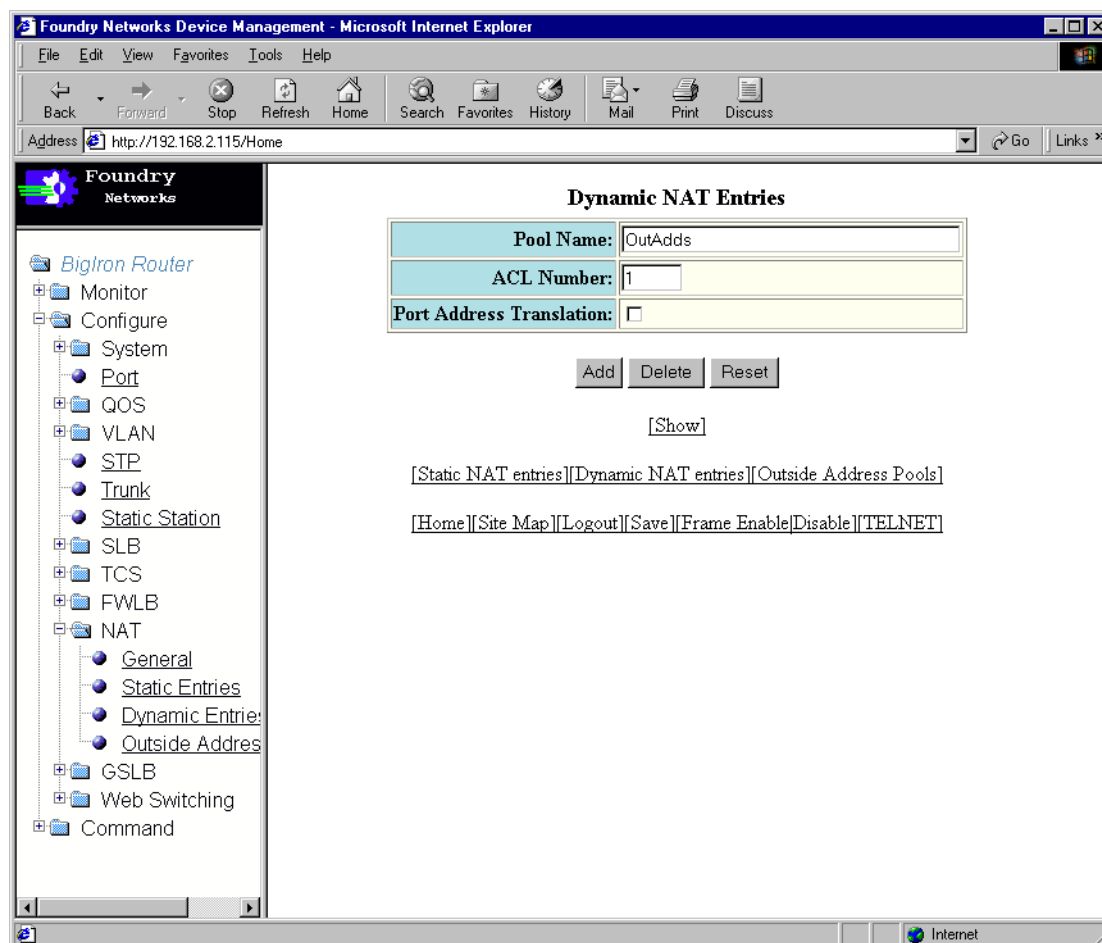
---

5. Enter the network mask for the address in the pool in the Mask field.
6. Select the Add button to add the pool to the device's running-config.
7. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.



## Associating the ACL with the Pool

1. Select the **Dynamic NAT Entries** link from the bottom of the panel or the Dynamic Entries link from the options menu to the left of the panel. The following panel will appear.



2. Enter the name of the pool you configured in "Configuring a Pool" on page 14-16 in the Pool Name field. The pool must already be configured.
3. Enter the number of the ACL you configured in "Associating the ACL with the Pool" on page 14-17 in the ACL Number field. The ACL must already be configured.
4. Optionally, click on the Port Address Translation checkbox to enable the feature. For a description of the feature, see "Port Address Translation" on page 14-2.
5. Select the Add button to add the addresses to the device's running-config.
6. Select the Save link at the bottom of the dialog, then select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.



---

# Appendix A

## Using Syslog

This appendix describes how to display Syslog messages and how to configure the Syslog facility, and lists the Syslog messages that a Foundry ServerIron can display during standard operation.

---

**NOTE:** This appendix does not list Syslog messages that can be displayed when a debug option is enabled. For information about Syslog messages that are displayed by a debug option, see the *Foundry Diagnostic Guide*.

---

---

**NOTE:** For information about Layer 2 Switch or Layer 3 Switch Syslog messages, see the “Using Syslog” appendix in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

---

## Overview

A Foundry device's software can write syslog messages to provide information at the following severity levels:

- Emergencies
- Alerts
- Critical
- Errors
- Warnings
- Notifications
- Informational
- Debugging

The device writes the messages to a local buffer that can hold up to 100 messages. You also can specify the IP address or host name of up to six SyslogD servers. When you specify a SyslogD server, the Foundry device writes the messages both to the system log and to the SyslogD server.

Using a SyslogD server ensures that the messages remain available even after a system reload. The Foundry device's local Syslog buffer is cleared during a system reload or reboot, but the Syslog messages sent to the SyslogD server remain on the server.

The SyslogD service on a Syslog server receives logging messages from applications on the local host or from devices such as a router or switch. SyslogD adds a time stamp to each received message and directs messages to a log file. Most Unix workstations come with SyslogD configured. Some third party vendor products also provide SyslogD running on NT.

SyslogD uses UDP port 514 and each SyslogD message thus is sent with destination port 514. Each SyslogD message is one line with SyslogD message format. The message is embedded in the text portion of the SyslogD format. There are several subfields in the format. Keywords are used to identify each subfield, and commas are delimiters. The subfield order is insensitive except that the text subfield should be the last field in the message. All the subfields are optional.

## Displaying Syslog Messages

To display the Syslog messages in the device's local buffer, enter the following command at any level of the CLI:

```
ServerIron> show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Buffer logging: level ACDMEINW, 3 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Static Log Buffer:
Dec 15 19:04:14:A:Fan 1, fan on right connector, failed

Dynamic Log Buffer (50 entries):
Dec 15 18:46:17:I:Interface ethernet4, state up
Dec 15 18:45:21:I:Bridge topology change, vlan 4095, interface 4, changed
state to forwarding
Dec 15 18:45:15:I:Warm start
```

## Additional Syslog Configuration Information

In general, the commands and options for changing Syslog parameters on a ServerIron are the same as for other Foundry products. For information, see the "Using Syslog" appendix in the *Foundry Switch and Router Installation and Basic Configuration Guide*.

## Syslog Messages

Table A.1 lists all of the Syslog messages. The messages are listed by message level, in the following order:

- Emergencies (none)
- Alerts
- Critical (none)
- Errors (none)
- Warnings
- Notifications
- Informational

- Debugging

**Table A.1: Foundry Syslog Messages**

Message Level	Message	Explanation
Alert	Power supply <num>, <location>, failed	<p>A power supply has failed.</p> <p>The &lt;num&gt; is the power supply number.</p> <p>The &lt;location&gt; describes where the failed power supply is in the chassis. The location can be one of the following:</p> <ul style="list-style-type: none"> <li>• In 4-slot Chassis devices: <ul style="list-style-type: none"> <li>• left side power supply</li> <li>• right side power supply</li> </ul> </li> <li>• In 8-slot Chassis devices: <ul style="list-style-type: none"> <li>• bottom power supply</li> <li>• middle bottom power supply</li> <li>• middle top power supply</li> <li>• top power supply</li> </ul> </li> <li>• In Stackable devices: <ul style="list-style-type: none"> <li>• power supply on right connector</li> <li>• power supply on left connector</li> </ul> </li> </ul>
Alert	Fan <num>, <location>, failed	<p>A fan has failed.</p> <p>The &lt;num&gt; is the power supply number.</p> <p>The &lt;location&gt; describes where the failed power supply is in the chassis. The location can be one of the following:</p> <ul style="list-style-type: none"> <li>• In 4-slot Chassis devices: <ul style="list-style-type: none"> <li>• left side panel, back fan</li> <li>• left side panel, front fan</li> <li>• rear/back panel, left fan</li> <li>• rear/back panel, right fan</li> </ul> </li> <li>• In 8-slot Chassis devices: <ul style="list-style-type: none"> <li>• rear/back panel, top fan</li> <li>• rear/back panel, bottom fan</li> <li>• top panel, fan</li> <li>• top panel, fan</li> </ul> </li> <li>• In Stackable devices: <ul style="list-style-type: none"> <li>• fan on right connector</li> <li>• fan on left connector</li> </ul> </li> </ul>

**Table A.1: Foundry Syslog Messages (Continued)**

Message Level	Message	Explanation
Alert	Management module at slot <slot-num> state changed from <module-state> to <module-state>.	<p>Indicates a state change in a management module.</p> <p>The &lt;slot-num&gt; indicates the chassis slot containing the module.</p> <p>The &lt;module-state&gt; can be one of the following:</p> <ul style="list-style-type: none"> <li>• active</li> <li>• standby</li> <li>• crashed</li> <li>• coming-up</li> <li>• unknown</li> </ul>
Alert	Temperature <degrees> C degrees, warning level <warn-degrees> C degrees, shutdown level <shutdown-degrees> C degrees	<p>Indicates an overtemperature condition on the active module.</p> <p>The &lt;degrees&gt; value indicates the temperature of the module.</p> <p>The &lt;warn-degrees&gt; value is the warning threshold temperature configured for the module.</p> <p>The &lt;shutdown-degrees&gt; value is the shutdown temperature configured for the module.</p>
Alert	<num-modules> modules and 1 power supply, need more power supply!!	<p>Indicates that the Chassis device needs more power supplies to run the modules in the chassis.</p> <p>The &lt;num-modules&gt; parameter indicates the number of modules in the chassis.</p>
Alert	Out of tcp send buffer at <application>	<p>Indicates that the TCP send buffer is exhausted.</p> <p>The &lt;application&gt; parameter is the application that caused the buffer overflow.</p>
Alert	Out of TCB memory at <application>	<p>Indicates that TCB memory is exhausted.</p> <p>The &lt;application&gt; parameter shows which application is out of TCB memory.</p>

Table A.1: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Warning	Locked address violation at interface e<portnum>, address <mac-address>	<p>Indicates that a port on which you have configured a lock-address filter received a packet that was dropped because the packet's source MAC address did not match an address learned by the port before the lock took effect.</p> <p>The e&lt;portnum&gt; is the port number.</p> <p>The &lt;mac-address&gt; is the MAC address that was denied by the address lock.</p> <p>Assuming that you configured the port to learn only the addresses that have valid access to the port, this message indicates a security violation.</p>
Warning	NTP server <ip-addr> failed to respond	<p>Indicates that a Simple Network Time Protocol (SNTP) server did not respond to the device's query for the current time.</p> <p>The &lt;ip-addr&gt; indicates the IP address of the SNTP server.</p>
Warning	Dup IP <ip-addr> detected, sent from MAC <mac-addr> interface <portnum>	<p>Indicates that the Foundry device received a packet from another device on the network with an IP address that is also configured on the Foundry device.</p> <p>The &lt;ip-addr&gt; is the duplicate IP address.</p> <p>The &lt;mac-addr&gt; is the MAC address of the device with the duplicate IP address.</p> <p>The &lt;portnum&gt; is the Foundry port that received the packet with the duplicate IP address. The address is the packet's source IP address.</p>
Warning	mac filter group denied packets on port <portnum> src macaddr <mac-addr>, <num> packets	<p>Indicates that a Layer 2 MAC filter group configured on a port has denied packets.</p> <p>The &lt;portnum&gt; is the port on which the packets were denied.</p> <p>The &lt;mac-addr&gt; is eth source MAC address of the denied packets.</p> <p>The &lt;num&gt; indicates how many packets matching the values above were dropped during the five-minute interval represented by the log entry.</p>

**Table A.1: Foundry Syslog Messages (Continued)**

Message Level	Message	Explanation
Warning	list <acl-num> denied <ip-proto> <src-ip-addr> (<src-tcp/udp-port>) (Ethernet <portnum> <mac-addr>) -> <dst-ip-addr> (<dst-tcp/udp-port>), <num> packets	<p>Indicates that an Access Control List (ACL) denied (dropped) packets.</p> <p>The &lt;acl-num&gt; indicates the ACL number. Numbers 1 – 99 indicate standard ACLs. Numbers 100 – 199 indicate extended ACLs.</p> <p>The &lt;ip-proto&gt; indicates the IP protocol of the denied packets.</p> <p>The &lt;src-ip-addr&gt; is the source IP address of the denied packets.</p> <p>the &lt;src-TCP/UDP-port&gt; is the source TCP or UDP port, if applicable, of the denied packets.</p> <p>The &lt;portnum&gt; indicates the port number on which the packet was denied.</p> <p>The &lt;mac-addr&gt; indicates the source MAC address of the denied packets.</p> <p>The &lt;dst-ip-addr&gt; indicates the destination IP address of the denied packets.</p> <p>The &lt;dst-tcp/udp-port&gt; indicates the destination TCP or UDP port number, if applicable, of the denied packets.</p> <p>The &lt;num&gt; indicates how many packets matching the values above were dropped during the five-minute interval represented by the log entry.</p>
Warning	firewall group <groupnum> become active	<p>Indicates that this ServerIron has become the active ServerIron in the high-availability (active-standby) FWLB configuration. (High-availability FWLB configurations also are called "IronClad" configurations.)</p> <p>The &lt;groupnum&gt; is the FWLB group ID, which normally is 2.</p>
Warning	firewall group <groupnum> become standby	<p>Indicates that this ServerIron has become the standby ServerIron in the high-availability (active-standby) FWLB configuration. (High-availability FWLB configurations also are called "IronClad" configurations.)</p> <p>The &lt;groupnum&gt; is the FWLB group ID, which normally is 2.</p>



Table A.1: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Warning	firewall path up target <target-ip-addr> nexthop <next-hop-ip-addr> path <path-id> port <portnum>	<p>Indicates that a firewall path has come up (and is therefore good).</p> <p>The &lt;target-ip-addr&gt; is the IP interface at the remote end of the path.</p> <p>The &lt;next-hop-ip-addr&gt; is the IP interface of the next hop in the path.</p> <p>The &lt;path-id&gt; is the ID you assigned to the path when you configured it.</p> <p>The &lt;portnum&gt; is the ServerIron port connected to the path's next hop.</p>
Warning	firewall path down target <target-ip-addr> nexthop <next-hop-ip-addr> path <path-id> port <portnum>	<p>Indicates that a firewall path has gone down (and is therefore unusable).</p> <p>The &lt;target-ip-addr&gt; is the IP interface at the remote end of the path.</p> <p>The &lt;next-hop-ip-addr&gt; is the IP interface of the next hop in the path.</p> <p>The &lt;path-id&gt; is the ID you assigned to the path when you configured it.</p> <p>The &lt;portnum&gt; is the ServerIron port connected to the path's next hop.</p>
Warning	HTTP match-list <matching-list> with simple pattern <string> Alert: bring server Down.	<p>Indicates that an HTTP content verification health check has matched a set of selection criteria specified in a <b>down simple</b> statement.</p> <p>When the selection criteria is found in the HTML file used for the health check, the ServerIron marks port 80 (HTTP) on the real server FAILED.</p> <p>&lt;matching-list&gt; is the name of the matching list whose selection criteria was matched.</p> <p>&lt;string&gt; is the selection criteria.</p>
Warning	HTTP match-list <policy-name> with simple pattern <string> Alert: bring server Up.	<p>Indicates that an HTTP content verification health check has matched a set of selection criteria specified in a <b>up simple</b> statement.</p> <p>When the selection criteria is found in the HTML file used for the health check, the ServerIron marks port 80 (HTTP) on the real server ACTIVE.</p> <p>&lt;policy-name&gt; is the name of the matching list whose selection criteria was matched.</p> <p>&lt;string&gt; is the selection criteria.</p>

Table A.1: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Warning	HTTP match-list <matching-list> with compound pattern1 <start> and pattern2 <end> Alert: bring server down and Extract message: <text-between-start-and-end-pattern>	<p>Indicates that an HTTP content verification health check has matched a set of selection criteria specified in a <b>down compound</b> statement.</p> <p>When the selection criteria is found in the HTML file used for the health check, the ServerIron marks port 80 (HTTP) on the real server FAILED.</p> <p>&lt;matching-list&gt; is the name of the matching list whose selection criteria was matched.</p> <p>&lt;start&gt; is the beginning of the selection criteria.</p> <p>&lt;end&gt; is the end of the selection criteria.</p>
Warning	HTTP match-list <matching-list> with compound pattern1 <start> and pattern2 <end> Alert: bring server up and Extract message: <text-between-start-and-end-pattern>	<p>Indicates that an HTTP content verification health check has matched a set of selection criteria specified in a <b>up compound</b> statement.</p> <p>When the selection criteria is found in the HTML file used for the health check, the ServerIron marks port 80 (HTTP) on the real server ACTIVE.</p> <p>&lt;matching-list&gt; is the name of the matching list whose selection criteria was matched.</p> <p>&lt;start&gt; is the beginning of the selection criteria.</p> <p>&lt;end&gt; is the end of the selection criteria.</p>
Warning	Port <TCP/UDP-portnum> on server <name>: <ip-addr>: Avg response time <num> exceeded lower threshold	<p>The application port on the real server did not respond within the warning threshold time.</p> <p>The &lt;TCP/UDP-portnum&gt; is the application port number.</p> <p>The &lt;name&gt; is the real server name.</p> <p>The &lt;ip-addr&gt; is the real server IP address.</p> <p>The &lt;num&gt; is the average number of milliseconds it was taking the application port to respond.</p>

Table A.1: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Warning	Port <TCP/UDP-portnum> on server <name>: <ip-addr>: Avg response time <num> exceeded upper threshold; Bringing down the port...	<p>The application port on the real server did not respond within the shutdown threshold time.</p> <p>The &lt;TCP/UDP-portnum&gt; is the application port number.</p> <p>The &lt;name&gt; is the real server name.</p> <p>The &lt;ip-addr&gt; is the real server IP address.</p> <p>The &lt;num&gt; is the average number of milliseconds it was taking the application port to respond.</p>
Notification	Module was inserted to slot <slot-num>	<p>Indicates that a module was inserted into a chassis slot.</p> <p>The &lt;slot-num&gt; is the number of the chassis slot into which the module was inserted.</p>
Notification	Module was removed from slot <slot-num>	<p>Indicates that a module was removed from a chassis slot.</p> <p>The &lt;slot-num&gt; is the number of the chassis slot from which the module was removed.</p>
Notification	L4 max connections <num> reached	<p>Indicates that the maximum number of connections supported by the ServerIron has been reached.</p> <p>The &lt;num&gt; indicates the number of connections.</p>
Notification	L4 TCP SYN limits <num> reached	<p>Indicates that the maximum number of connections per second allowed by the ServerIron has been reached.</p> <p>The &lt;num&gt; indicates the number of connections.</p>
Notification	L4 server <ip-addr> <name> max connections <num> reached	<p>Indicates that the maximum number of connections allowed on a real server has been reached.</p> <p>The &lt;ip-addr&gt; is the real server's IP address.</p> <p>The &lt;name&gt; is the name of the real server.</p> <p>The &lt;num&gt; indicates the number of connections.</p>
Notification	L4 begin-holddown source-ip <src-ip-addr> dest-ip <dst-ip-addr>	<p>Indicates that the ServerIron's SYN attack prevention feature is "holding down" the specified source and destination IP address pair, which means the ServerIron is not sending these packets to any servers.</p>

**Table A.1: Foundry Syslog Messages (Continued)**

Message Level	Message	Explanation
Notification	L4 server <ip-addr> <name> is up	<p>Indicates that a real server or cache server has come up.</p> <p>The &lt;ip-addr&gt; is the server's IP address.</p> <p>The &lt;name&gt; is the name of the server.</p>
Notification	<p>L4 server &lt;ip-addr&gt; &lt;name&gt; is down due to &lt;reason&gt;</p> <p><b>Note:</b> The "due to &lt;reason&gt;" portion is supported only in software release 07.2.25 and later 07.2.x releases.</p>	<p>Indicates that a real server or cache server has gone down.</p> <p>The &lt;ip-addr&gt; is the server's IP address.</p> <p>The &lt;name&gt; is the name of the server.</p> <p>The &lt;reason&gt; is the reason the ServerIron changed the port's state to down. The &lt;reason&gt; can be one of the following:</p> <ul style="list-style-type: none"> <li>healthck – The port failed a health check. This applies to standard health checks and Boolean health checks.</li> <li>reassign – The reassign threshold was reached.</li> <li>server-down – The server failed the Layer 3 health check when you bound the real server to the VIP.</li> <li>MAC-delete – The server's MAC address was deleted from the ServerIron MAC table.</li> <li>graceful-shutdown – The server was gracefully shut down.</li> <li>mp-port-state-change – The port was brought down on the WSM CPU managing the real server, in response to a message from the MP CPU that the port is down.</li> </ul> <p><b>Note:</b> This value applies only to the ServerIron 400 and ServerIron 800.</p> <ul style="list-style-type: none"> <li>other – The port was brought down by another application (by something other than the ServerIron.)</li> <li>unknown – The port was brought down by a reason other than one of those listed above.</li> </ul>
Notification	L4 server <ip-addr> <name> TCP port <tcp-port-num> is up	<p>Indicates that a real server's or cache server's TCP port has come up.</p> <p>The &lt;ip-addr&gt; is the server's IP address.</p> <p>The &lt;name&gt; is the name of the server.</p> <p>The &lt;tcp-port-num&gt; is the TCP port number.</p>

Table A.1: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Notification	L4 server <ip-addr> <name> TCP port <tcp-port-num> is down	Indicates that a real server's or cache server's TCP port has gone down.  The <ip-addr> is the server's IP address.  The <name> is the name of the server.  The <tcp-port-num> is the TCP port number.
Notification	L4 switch changed state from active to standby	The ServerIron is an active-standby configuration and has changed from the active to the standby state.
Notification	L4 switch changed state from standby to active	The ServerIron is an active-standby configuration and has changed from the standby to the active state.
Notification	L4 gslb connection to site <name> SI <ip-addr> <name> is up	The GSLB protocol connection from this GSLB ServerIron to a remote site ServerIron has come up.  The first <name> the site name.  The <ip-addr> and <name> are the site ServerIron's management IP address and name.
Notification	L4 gslb connection to site <name> SI <ip-addr> <name> is down	The GSLB protocol connection from this GSLB ServerIron to a remote site ServerIron went down.  The first <name> the site name.  The <ip-addr> and <name> are the site ServerIron's management IP address and name.
Notification	L4 gslb connection to gslb SI <ip-addr> is up	The GSLB protocol connection from this site ServerIron to a remote GSLB ServerIron has come up.  The <ip-addr> is the GSLB ServerIron's management IP address.
Notification	L4 gslb connection to gslb SI <ip-addr> is down	The GSLB protocol connection from this site ServerIron to a remote GSLB ServerIron has gone down.  The <ip-addr> is the GSLB ServerIron's management IP address.
Notification	L4 gslb health-check <ip-addr> of <zone> status changed to up	The IP address belonging to a domain name for which the ServerIron is providing GSLB has come up.  The <ip-addr> is the IP address in the DNS reply.  The <zone> is the zone name.

**Table A.1: Foundry Syslog Messages (Continued)**

Message Level	Message	Explanation
Notification	L4 gslb health-check <ip-addr> of <zone> status changed to down	<p>The IP address belonging to a domain name for which the ServerIron is providing GSLB has gone down.</p> <p>The &lt;ip-addr&gt; is the IP address in the DNS reply.</p> <p>The &lt;zone&gt; is the zone name.</p>
Notification	L4 gslb health-check <ip-addr> of <zone> port <tcp/udp-port> is up	<p>An application port in a domain on the site IP address passed its Layer 4 TCP or UDP health check.</p> <p>The &lt;ip-addr&gt; is the IP address in the DNS reply.</p> <p>The &lt;zone&gt; is the zone name.</p> <p>The &lt;tcp/udp-port&gt; is the application port.</p>
Notification	L4 gslb health-check <ip-addr> of <zone> port <tcp/udp-port> is down	<p>An application port in a domain on the site IP address failed its Layer 4 TCP or UDP health check.</p> <p>The &lt;ip-addr&gt; is the IP address in the DNS reply.</p> <p>The &lt;zone&gt; is the zone name.</p> <p>The &lt;tcp/udp-port&gt; is the application port.</p>
Informational	Cold start	The device has been powered on.
Informational	Warm start	The system software (flash code) has been reloaded.
Informational	<user-name> login to USER EXEC mode	<p>A user has logged into the USER EXEC mode of the CLI.</p> <p>The &lt;user-name&gt; is the user name.</p>
Informational	<user-name> logout from USER EXEC mode	<p>A user has logged out of the USER EXEC mode of the CLI.</p> <p>The &lt;user-name&gt; is the user name.</p>
Informational	<user-name> login to PRIVILEGED mode	<p>A user has logged into the Privileged EXEC mode of the CLI.</p> <p>The &lt;user-name&gt; is the user name.</p>
Informational	<user-name> logout from PRIVILEGED mode	<p>A user has logged out of Privileged EXEC mode of the CLI.</p> <p>The &lt;user-name&gt; is the user name.</p>
Informational	SNMP Auth. failure, intruder IP: <ip-addr>	<p>A user has tried to open a management session with the device using an invalid SNMP community string.</p> <p>The &lt;ip-addr&gt; is the IP address of the host that sent the invalid community string.</p>

Table A.1: Foundry Syslog Messages (Continued)

Message Level	Message	Explanation
Informational	Interface <portnum>, state up	A port has come up. The <portnum> is the port number.
Informational	Interface <portnum>, state down	A port has gone down. The <portnum> is the port number.
Informational	Bridge root changed, vlan <vlan-id>, new root ID <root-id>, root interface <portnum>	A Spanning Tree Protocol (STP) topology change has occurred.  The <vlan-id> is the ID of the VLAN in which the STP topology change occurred.  The <root-id> is the STP bridge root ID.  The <portnum> is the number of the port connected to the new root bridge.
Informational	Bridge is new root, vlan <vlan-id>, root ID <root-id>	A Spanning Tree Protocol (STP) topology change has occurred, resulting in the Foundry device becoming the root bridge.  The <vlan-id> is the ID of the VLAN in which the STP topology change occurred.  The <root-id> is the STP bridge root ID.
Informational	Bridge topology change, vlan <vlan-id>, interface <portnum>, changed state to <stp-state>	A Spanning Tree Protocol (STP) topology change has occurred on a port.  The <vlan-id> is the ID of the VLAN in which the STP topology change occurred.  The <portnum> is the port number.  The <stp-state> is the new STP state and can be one of the following: <ul style="list-style-type: none"> <li>• disabled</li> <li>• blocking</li> <li>• listening</li> <li>• learning</li> <li>• forwarding</li> <li>• unknown</li> </ul>





---

# Appendix B

## Network Monitoring

This chapter provides a general overview of monitoring tools supported on the ServerIron. It includes the following topics:

- Configuring RMON
- Monitoring Layer 4 Statistics
- Viewing System Information
- Viewing Configuration Information
- Viewing Port Statistics
- Viewing STP Statistics

### Configuring RMON

All Foundry Networks switches and switching routers come standard with an RMON agent that supports the following groups. The group numbers come from the RMON specification (RFC 1757).

- Statistics (RMON Group 1)
- History (RMON Group 2)
- Alarms (RMON Group 3)
- Events (RMON Group 9)

The CLI allows you to make configuration changes to the control data for these groups, but you need a separate RMON application to view and display the data graphically.

#### Statistics (RMON Group 1)

Count information on multicast and broadcast packets, total packets sent, undersized and oversized packets, CRC alignment errors, jabbers, collision, fragments and dropped events is collected for each port on the ServerIron.

No configuration is required to activate collection of statistics for the switch or router. This activity is by default automatically activated at system start-up.

#### *USING THE CLI*

You can view a textual summary of the statistics for all ports by entering the following CLI command:

```
ServerIron(config)# show rmon statistics
```

**Syntax:** show rmon statistics <portnum>

**NOTE:** The number of entries in a RMON statistics table directly corresponds to the number of ports on a system. For example, if the system is a 26 port device, there will be 26 entries in the statistics display.

To see RMON statistics for an individual port only, enter the following command noting a specific port entry number: **show rmon statistics** <entry-number>.

---

#### *USING THE WEB MANAGEMENT INTERFACE*

To view the RMON statistics for the system:

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to Port in the tree view to expand the list of Port option links.
4. Click on the Statistic link to display the Port Statistic table.
5. Click on the RMON Ethernet Statistics link to display the RMON Ethernet Statistics table.

### **History (RMON Group 2)**

All active ports by default will generate two history control data entries per active Foundry switch port or router interface. An active port is defined as one with a link up. If the link goes down the two entries are automatically be deleted.

Two history entries are generated for each device:

- a sampling of statistics every 30 seconds
- a sampling of statistics every 30 minutes

The history data can be accessed and displayed using any of the popular RMON applications

#### *USING THE CLI*

A sample RMON history command and its syntax is shown below:

```
ServerIron(config)# rmon history 1 interface 1 buckets 10 interval 10 owner nyc02
```

**Syntax:** rmon history <entry-number> interface <portnum> buckets <number> interval <sampling-interval> owner <text-string>

You can modify the sampling interval and the buckets (number of entries saved before overwrite) using the CLI. In the above example, owner refers to the RMON station that will request the information.

---

**NOTE:** To review the control data entry for each port or interface, enter the **show rmon history** command.

---

#### *USING THE WEB MANAGEMENT INTERFACE*

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to Port in the tree view to expand the list of Port option links.
4. Click on the Statistics link to display the Port Statistic table.
5. Click on the History link to display the RMON Ethernet History table.

### **Alarm (RMON Group 3)**

Alarm is designed to monitor configured thresholds for any SNMP integer, time tick, gauge or counter MIB object. Using the CLI, you can define what MIB objects are monitored, the type of thresholds that are monitored (falling, rising or both), the value of those thresholds, and the sample type (absolute or delta).

An alarm event is reported each time that a threshold is exceeded. The alarm entry also indicates the action (event) to be taken if the threshold be exceeded.

#### USING THE CLI

A sample CLI alarm entry and its syntax is shown below:

```
ServerIron(config)# rmon alarm 1 ifInOctets.6 10 delta rising-threshold 100 1
falling threshold 50 1 owner nyc02
```

**Syntax:** rmon alarm <entry-number> <MIB-object.interface-num> <sampling-time> <sample-type> <threshold-type> <threshold-value> <event-number> <threshold-type> <threshold-value> <event-number> owner <text-string>

#### USING THE WEB MANAGEMENT INTERFACE

This display is not supported on the Web management interface.

## Event (RMON Group 9)

There are two elements to the Event Group—the *event control table* and the *event log table*.

The event control table defines the action to be taken when an alarm is reported. Defined events can be found by entering the CLI command, **show rmon event**. The Event Log Table collects and stores reported events for retrieval by an RMON application.

#### USING THE CLI

A sample entry and syntax of the event control table is shown below:

```
ServerIron(config)# rmon event 1 description 'testing a longer string' log-and-trap
public owner nyc02
```

**Syntax:** rmon event <event-entry> description <text-string> log | trap | log-and-trap owner <rmon-station>

#### USING THE WEB MANAGEMENT INTERFACE

This display is not supported on the Web management interface.

## Monitoring Layer 4 Statistics

The ServerIron has an RMON-like monitoring function for gathering and recording Layer 4 statistics from real servers and virtual servers. Two groups are supported:

- Layer 4 Statistics group
- Layer 4 History group

You configure the control data for the Layer 4 History group. The data can be viewed using the Web management interface or a separate NMS application. Data is gathered continuously, even when the ServerIron is not being polled by an NMS application.

## Layer 4 Statistics Group

The Layer 4 Statistics group contains information about real and virtual servers. This is the same information that is displayed by the **show server real** and **show server virtual** CLI commands. For example:

```
ServerIron(config)# show server virtual
Server Name: aaa                IP : 1.2.3.55                :    1
Status: enabled  Predictor: least-conn  TotConn: 0
Dynamic: No      HTTP redirect: disabled
                        Intercept: No
ACL: id =    0
Sym: group =   1 state =   1 priority =   0 keep =   0
  Activates =   0, Inactive= 0
Port    State    Sticky  Concur  Proxy          CurConn    TotConn    PeakConn

http    enabled   NO      NO      NO              0           0           0
default enabled   NO      NO      NO              0           0           0
```

```
ServerIron(config) show server real

Name : bbb                                Mac-addr: Unknown
IP:1.2.3.66      Range:1    State:Enabled      Max-conn:1000000
Least-con Wt:0   Resp-time Wt:0
```

Port	State	Ms	CurConn	TotConn	Rx-pkts	Tx-pkts	Rx-octet	Tx-octet	Reas
----	-----	--	-----	-----	-----	-----	-----	-----	----
http	unbnd	0 0	0	0	0	0	0	0	0
default	unbnd	0 0	0	0	0	0	0	0	0
Server	Total		0	0	0	0	0	0	0

Information collected in the Layer 4 Statistics group includes:

- Rx-pkts**     The number of packets the ServerIron has received from the server.
- Tx-pkts**     The number of packets the ServerIron has sent to the server.
- CurConn**    The number of client connections currently on the server. A connection consists of two sessions, the client-to-server session and the server-to-client session.
- PeakConn**    The highest number of connections the VIP has had at the same time.

## Layer 4 History Group

The Layer 4 History group consists of the following tables:

- historyControlTable
- realServerHistoryTable
- virtualServerHistoryTable
- realServerPortHistoryTable
- virtualServerPortHistoryTable

The historyControlTable contains control data for the history group, including the history list index number, monitored server and port name, allocated buckets, sampling interval, and owner. This data is configured by creating a **history list** and then binding the history list to a real server, virtual server, or a port on a real or virtual server. The other tables contain statistical data gathered using information in the historyControlTable.

## Creating a History List

To create a history list, enter commands such as the following

```
ServerIron(config)# server monitor
ServerIron(config-slb-mon)# history 1 buckets 5 interval 30 owner rkwong
ServerIron(config-slb-mon)# history 2 buckets 10 interval 30 owner fdry
```

**Syntax:** server monitor

**Syntax:** history <entry-number> buckets <number> interval <sampling-interval> owner <text-string>

The **server monitor** command enters the Layer 4 monitor CLI level.

The **history** commands configure the history lists. In the example above, two history lists are configured. You can set the following parameters in a history list:

- <entry-number> Is the index number for the history list. This can be a number from 1 – 100.
- buckets <number> Is the number of rows allocated to a data table for this history list. This can be a number from 1 – 65535. This number of samples are stored in the data table. For example, if you specify 10 buckets, the most recent 10 samples are stored in the data table.
- interval <sampling-interval> Is the sampling interval in seconds. The sampling interval can be from 1 – 3600 seconds.
- owner <text-string> Specifies the owner of the history list.

## Binding the History List to the Server

After you create the history list, you bind it to a real server, virtual server, or to a port on a real or virtual server. For example, to bind the two history lists created above to a real server, enter commands such as the following:

```
ServerIron(config)# server real aaa
ServerIron(config-rs-aaa)# history-group 1 2
```

To bind the history lists to port 80 (HTTP) on real server aaa, enter commands such as the following

```
ServerIron(config)# server real aaa
ServerIron(config-rs-aaa)# port http history-group 1 2
```

To bind the history lists to a virtual server, enter commands such as the following:

```
ServerIron(config)# server virtual bbb
ServerIron(config-vs-bbb)# history-group 1 2
```

To bind the history lists to port 80 (HTTP) on virtual server bbb, enter commands such as the following

```
ServerIron(config)# server virtual bbb
ServerIron(config-vs-bbb)# port http history-group 1 2
```

**Syntax:** history-group <entry-numbers>

The **history-group** command binds the history lists to the real server. You can bind up to 8 history lists to a server or port in this way.

Information you specify in a history list is added to the historyControlTable. The ServerIron adds entries to the data tables based on information in the historyControlTable. For example, after the two history lists configured above are bound to real server aaa, the realServerHistoryTable would contain data similar to the following:

Entry Number	Sampling Index	Interval Start	Rx-pkts	Tx-pkts	CurConn	PeakConn
1	33400	11 days 14:30:01				
1	33401	11 days 14:30:31				

Entry Number	Sampling Index	Interval Start	Rx-pkts	Tx-pkts	CurConn	PeakConn
1	33402	11 days 14:31:01				
1	33403	11 days 14:31:31				
1	33404	11 days 14:32:01				
2	1	0 days 00:00:01				
2	2	0 days 00:00:31				
2	3	0 days 00:01:01				
2	4	0 days 00:01:31				
2	5	0 days 00:02:01				
2	6	0 days 00:02:31				
2	7	0 days 00:03:01				
2	8	0 days 00:03:31				
2	9	0 days 00:04:01				
2	10	0 days 00:04:31				

For each index entry, there are a number of rows equal to the number of buckets specified in the history list. Each time the ServerIron takes a sample, the data is stored in one of the rows allocated to the index entry. For example, for index entry 2, the ServerIron takes a sample once every 30 seconds. Each sample is stored in a row of the realServerHistoryTable, and the most recent 10 rows (10 buckets) are retained.

#### USING THE WEB MANAGEMENT INTERFACE

Creating history lists and binding them to servers and ports is supported only through the CLI. However, you can use the Web management interface to view the statistics collected by the ServerIron. To do so:

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to SLB in the tree view to expand the list of SLB option links.
4. Click on the plus sign next to History in the tree view to expand the list of statistics history option links.
5. Click on the link for the information you want to view.

## Viewing System Information

You can access software and hardware specifics for a ServerIron.

#### USING THE CLI

To view the software and hardware details for the system, enter the **show version** command:

```
ServerIron# show version
```

**Syntax:** show version

#### USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.

2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the [Device](#) link to display the Device Information panel.

## Viewing Configuration Information

You can view a variety of configuration details and statistics with the show option. The **show** option provides a convenient way to check configuration changes before saving them to flash.

The show options available will vary by configuration level.

### [USING THE CLI](#)

To determine the available show commands for the system or a specific level of the CLI, enter the following command:

```
ServerIron(config)# # show ?
```

**Syntax:** show <option>

You also can enter **show** at the command prompt, then press the TAB key.

---

**NOTE:** For a complete summary of all available **show...** CLI commands and their displays, see the *Foundry Switch and Router Command Line Interface Reference*.

---

### [USING THE WEB MANAGEMENT INTERFACE](#)

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. If needed, click on the plus sign next to a subcategory to display the monitoring links for that category.
4. Click on the link for the information you want to view.

## Viewing Port Statistics

Port statistics are polled by default every 10 seconds.

### [USING THE CLI](#)

You can view statistics for ports by entering the following **show** commands:

- **show interfaces**
- **show configuration**

### [USING THE WEB MANAGEMENT INTERFACE](#)

To view the port statistics for all ports on the ServerIron:

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Click on the plus sign next to Port to expand the list of port monitoring options.
4. Select the [Statistic](#) link.

## Viewing STP Statistics

You can view a summary of STP statistics on the ServerIron. STP statistics are by default polled every 10 seconds.

To modify this polling rate (when using the Web management interface), select the [Preferences](#) link from the main menu, and modify the STP field. You can disable polling by setting the field to zero.

#### [USING THE CLI](#)

To view spanning tree statistics, enter the **show span** command. To view STP statistics for a VLAN, enter the **span vlan** command.

#### [USING THE WEB MANAGEMENT INTERFACE](#)

1. Log on to the device using a valid user name and password for read-only or read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Monitor in the tree view to expand the list of monitoring options.
3. Select the [STP](#) link.

## Clearing Statistics

You can clear statistics for many parameters with the clear option.

#### [USING THE CLI](#)

To determine the available **clear** commands for the system, enter the following command:

```
ServerIron(config)# # clear ?
```

**Syntax:** clear <option>

You also can enter “clear” at the command prompt, then press the TAB key.

For a complete summary of all available **clear...** CLI commands and their displays, see the *Foundry Switch and Router Command Line Interface Reference*.

---

**NOTE:** Clear commands are found at the Privileged EXEC level.

---

#### [USING THE WEB MANAGEMENT INTERFACE](#)

You can clear statistics by doing the following:

1. Log on to the device using a valid user name and password for read-write access. The System configuration dialog is displayed.
2. Click on the plus sign next to Command in the tree view to expand the list of command options.
3. Click on the [Clear](#) link to display the Clear panel.
4. Select all items to be cleared.
5. Click Apply.



---

## Appendix C

### HTTP Status Codes

The ServerIron can perform HTTP health checks for TCS and SLB implementations. The ServerIron makes a determination about the health of the HTTP service on a real server based on its response to an HTTP GET or HEAD request.

- If the real server responds before the HTTP keepalive interval expires, the ServerIron assumes that the HTTP service on that real server is alright and marks the service **ACTIVE**.
- However, if the real server responds with an HTTP reply status code less than 200 or greater than 299 (for SLB) or less than 200 or greater than 499 (for TCS), the ServerIron marks the HTTP service on the real server **FAILED**.
- If the real server does not respond, the ServerIron retries the request up to the number of times specified by the HTTP retries parameter. If the real server's HTTP service still does not respond, the ServerIron marks the service **FAILED** for that server.

---

**NOTE:** You can change the status code ranges. See "Modifying the HTTP Keepalive Method, Value, and Status Codes" on page 12-32.

---

Table C.1 on page C-1 lists the standard HTTP status codes.

**Table C.1: HTTP Status Codes**

Code	Meaning	ServerIron marks HTTP FAILED	
		TCS	SLB
<b>100 – 199</b>	Informational		
100	Continue	x	x
101	Switching protocols (not used yet, but defined)	x	x
<b>200 – 299</b>	Success		
200	OK		
201	Created		
202	Accepted		

**Table C.1: HTTP Status Codes (Continued)**

Code	Meaning	ServerIron marks HTTP FAILED	
		TCS	SLB
203	Non-Authoritative information		
204	No Content		
205	Reset content		
206	Partial content		
<b>300 – 399</b>	Redirection		
300	Multiple choices		x
301	Moved Permanently		x
302	Moved Temporarily		x
303	See Other		x
304	Not Modified		x
305	Use Proxy		x
<b>400 – 499</b>	Client Error		
400	Bad Request		x
401	Unauthorized		x
402	Payment Required		x
403	Forbidden		x
404	Not Found		x
405	Method not allowed		x
406	Not Acceptable		x
407	Proxy Authentication Required		x
408	Request timeout		x
409	Conflict		x
410	Gone		x
411	Length Required		x
412	Precondition Failed		x
413	Request entity too large		x
414	Request URI too large		x
415	Unsupported media type		x
<b>500 – 599</b>	Server Error <sup>a</sup>		
500	Internal Server Error	x	x

Table C.1: HTTP Status Codes (Continued)

Code	Meaning	ServerIron marks HTTP FAILED	
		TCS	SLB
501	Not Implemented	x	x
502	Bad Gateway	x	x
503	Service Unavailable	x	x
504	Gateway time-out	x	x
505	HTTP version not supported - or - extension-code	x	x

a. These error codes refer to errors on the HTTP server, not on the ServerIron.



## Numerics

802.1q 2-40

## A

Access

SNMP

configuring 3-11

Web management interface 3-11

access control 2-10

access, CLI 3-11

address-lock filter 2-41

age

TCP

changing 12-59

TCP/UDP port

overriding 12-22, 12-27

UDP

changing 12-59

Alarm

RMON Group 3 B-2

application group 2-23

configuring 6-56

SLB application example 6-104

assigning

interface to a cache group 10-10

IP addresses to web cache server 10-7

## B

binding 6-20

displaying information 6-92

## C

cache

policy-based cache failover 2-36

cache group

assigning an interface 10-10

disabling 10-10

interface 10-11

cache route optimization (CRO) 2-36

application example 10-35

enabling 10-16

cache server

adding to cache group 10-8

configuring 10-7

disabling 10-10

shutting down 10-31

CFO 2-36

Chassis

poll interval 4-14

CLI 2-10

CLI access 3-11

command-line interface (CLI) 2-10

Community string

configuring 3-11

encryption 3-12

concurrent connections 2-24

connection

limiting 6-45, 10-18, 12-58

contents

package 3-1

conventions

manual 1-1

CRO 2-36

## D

desktop installation 3-6

destination NAT

enabling 10-19

direct cache-server return

application example 10-39

enabling 10-19

see FastCache

Direct Server Return

see SwitchBack

DNS resolver 2-12

dynamic configuration 2-11

Dynamic Host Configuration Protocol (DHCP) assist 2-41

## E

- Email Access 1-9
- enabling transparent cache switching 10-4
- Encryption
  - SNMP community string 3-12
- Event
  - RMON Group 9 B-3

## F

- failover
  - cache 2-36
- FastCache
- features
  - hardware 2-1
  - Layer 2 2-38
  - Layer 4-7 2-1
  - overview 2-5
- File synchronization
  - redundant management module 4-19
- filter
  - address-lock 2-41
  - MAC 2-40
  - SLB
    - configuring 6-67
  - TCS 2-17, 10-24
- firewall load balancing 2-37
- flash code
  - version 2-4
- forced server shutdown 2-21, 6-69
  - enabling 6-34, 10-17

## G

- getting help 1-9
- grounding 3-2

## H

- hardware features 2-1
- hash mask
  - TCS 10-12
- health check
  - DNS
    - configuring 12-38
  - enabling 12-23
  - HTTP
    - configuring 12-32
  - keepalive interval and retries
    - modifying 12-23
  - Layer 3 2-19
  - Layer 4 2-19
  - RADIUS
    - configuring 12-38
- help
  - getting 1-9
- History
  - RMON Group 2 B-2
- host range 2-25
  - application example 6-101
- hot standby
  - viewing information 5-12

- hot standby redundancy
  - configuring 5-1
- hot standby redundancy 2-21
- HTTP redirect 2-29
  - application example 6-113
- HTTP status codes C-1

## I

- ICMP message feature 2-19, 6-29
  - enabling 6-29
- Installation
  - location and clearance 3-3
- installation
  - desktop 3-6
- interface
  - cache group 10-11
- IP address
  - source NAT 6-30
- IP multicast containment 2-41
- IP/RIP
  - redistribution
    - enabling 13-12

## L

- Layer 2 features 2-38
- Layer 3
  - basic services 2-14
  - health check 2-19
- Layer 4
  - health check 2-19
- Layer 4-7
  - features 2-1
- least-connections 6-24
- LED behavior 3-3
- LEDs 2-3
  - redundant management module 4-18
  - WSMM 4-27
- link-level redundancy 2-27
- load-balancing metric 2-23
  - changing 6-24

## M

- MAC
  - static entry 2-39
- MAC filter 2-40
- MAC switching 2-38
- manual nomenclature 1-1
- many-to-one port binding 2-29
  - application example 6-98
- message buffer 2-13
- modifying
  - cache server default settings 10-17
- Module
  - redundant management
    - configuring 4-16
    - default active module 4-17
    - file synchronization 4-19

---

- status 4-18
- WSMM
  - displaying information 4-25
  - status 4-26, 4-30
- multicast 2-41
- N**
- NAT 2-15
- network address translation (NAT) 2-15
- network connections
  - troubleshooting 3-9
- O**
- on-going CLI access 3-11
- P**
- package contents 3-1
- ping 2-14
- policy-based cache failover (CFO) 2-36
  - application example 10-41
  - configuring 10-14
- policy-based cache switching
  - enabling 10-22
- policy-based caching
  - application example 10-38
- Poll interval 4-14
- Port
  - statistics B-7
- port binding
  - many-to-one 2-29
- port mirroring 2-14
- port priority
  - TCP/UDP port
    - priority 10-23
- port profile
  - profile
    - TCP/UDP port 2-18, 12-21
- port-based VLAN 2-40
- Power Cord
  - caution 3-2
- power cord 3-3
- predictor 2-23
  - changing 6-24
- priority 10-23
- Proxy problem
  - Web management access 2-8

- Q**
- QoS 2-14
- Quality of Service (QoS) 2-14

- R**
- Read-write community string
  - no default 3-11
- real server
  - configuring 6-12
- reassign threshold 12-19, 12-29
- Redistribution

- IP/RIP
  - enabling 13-12
- redundancy
  - hot standby 2-21
    - configuring 5-1
- Redundant management module
  - configuring 4-16
  - default active module 4-17
  - file synchronization 4-19
  - status 4-18
- related publications 1-2
- reload 2-11
- remote server 2-25
- remote servers
  - SLB application example 6-110
- reverse NAT
  - disabling 6-33
- RMON 2-13, B-1
- round robin 6-25
- router port 6-27
- S**
- scheduled system reload 2-11
- Secure Socket Layer (SSL) 2-33
- server
  - real
    - configuring 6-12
    - shutting down 6-69
  - virtual
    - configuring 6-17
- Server Load Balancing (SLB) 2-21
- session
  - limiting 10-18, 12-58
  - statistics
    - displaying 6-93
- SLB 2-21
  - application examples 6-96, 13-20
  - binding
    - displaying information 6-92
  - configuring 6-1
  - displaying global configuration information 6-70
  - real server
    - displaying information 6-74
  - remote server 2-25
  - session statistics
    - displaying 6-93
  - Symmetric 2-26
  - traffic statistics
    - displaying 6-94
  - viewing information 6-69
  - virtual server
    - displaying information 6-82
- SNMP
  - community string
    - configuring 3-11
    - encryption 3-12
- SNMP trap 2-20
- SNTP 2-12

- soft reboot 2-11
- Software
  - synchronization
    - redundant management module 4-19
  - WSMM version 4-24
- source NAT 6-30
  - SLB application example 6-107
- Spanning Tree Protocol (STP) 2-39
- stateful TCS 2-35
- static MAC entry 2-39
- Statistics
  - clearing B-8
  - port B-7
  - RMON Group 1 B-1
  - STP B-7
- statistics
  - RMON 2-13
  - SLB session
    - displaying 6-93
  - SLB traffic
    - displaying 6-94
- status codes
  - HTTP C-1
- sticky age 6-34, 6-105
- sticky connection 2-24
- STP
  - statistics B-7
- SwitchBack 2-27
  - application example 7-15
- Switchover
  - redundant management module 4-15
  - Syslog messages 4-19
- Symmetric SLB 2-26
  - application example 7-1
- Syslog 2-13
  - redundant management switchover 4-19
  - temperature 4-12
- System
  - displaying information B-6
- system
  - unpacking 3-1
- system time 2-12

## T

- tagging
  - VLAN 2-40
- TCP age
  - changing 12-59
- TCP SYN
  - limiting 6-27
- TCP/UDP port
  - application group
    - configuring 6-56
  - health check
    - enabling 12-23
  - local parameters
    - changing 12-31
  - overriding TCP or UDP age 12-22, 12-27

- profile 2-18, 12-21
- TCP/UDP ports
  - binding 6-20
- TCP-SYN ACK
  - limiting 12-19, 12-29
- TCS 2-34
  - application example 10-3
  - application examples 10-32
  - configuring 10-1
  - enabling 10-4
  - filter 2-17
  - hash mask 10-12
  - stateful 2-35
  - viewing information 10-30
- Telephone Access 1-9
- Telnet
  - ServerIron management 2-11
- Temperature
  - displaying 4-11
  - poll interval 4-14
  - sensor 4-11
    - changing warning and shutdown levels 4-13
  - Syslog 4-12
- temperature 3-2
- TFTP
  - flash upgrade 2-12
- time 2-12
- traffic
  - statistics
    - displaying 6-94
- Transparent Cache Switching
  - configuration notes 10-3
  - enabling 10-4
- trap
  - SNMP 2-20
- troubleshooting network connections 3-9
- trunk group 2-40

## U

- UDP age
  - changing 12-59
- unlimited VIPs 2-25
  - application example 6-101
- unpacking a system 3-1

## V

- Version
  - WSMM software 4-24
- virtual server
  - configuring 6-17
- VLAN
  - port-based 2-40
- VLAN tagging 2-40

## W

- Web Access 1-9
- web cache server
  - assigning IP addresses 10-7



---

- web hosting
  - SLB application example 6-97, 6-104
- Web management interface 2-8
  - access 3-11
  - proxy problem 2-8
- weight 6-48, 10-18
- weighted percentage 6-25
- WSMM
  - displaying information 4-24
  - LEDs 4-27
  - module
    - displaying information 4-25
    - status 4-26, 4-30

