Cisco Content Services Switch Redundancy Configuration Guide

Software Version 8.10
November 2005

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Text Part Number: OL-8243-01

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco Content Services Switch Redundancy Configuration Guide

CONTENTS

FIGURES

TABLES

Preface

This guide provides instructions for configuring the redundancy features of the Cisco 11500 Series Content Services Switches (CSS). Information in this guide applies to all CSS models except where noted.

The CSS software is available in a Standard or optional Enhanced feature set. The Enhanced feature set contains all of the Standard feature set and also includes Network Address Translation (NAT) Peering, Domain Name Service (DNS), Demand-Based Content Replication (Dynamic Hot Content Overflow), Content Staging and Replication, and Network Proximity DNS. Proximity Database and Secure Management, which includes Secure Shell Host and SSL strong encryption, are optional features.

This preface contains the following major sections:

- Audience
- How to Use This Guide
- Related Documentation
- Symbols and Conventions
- Obtaining Documentation
- Documentation Feedback
- Cisco Product Security Overview
- Obtaining Technical Assistance
- Obtaining Additional Publications and Information

Audience

This guide is intended for the following trained and qualified service personnel who are responsible for configuring the CSS:

- Web master
- System administrator
- System operator

How to Use This Guide

This guide is organized as follows:

Chapter	Description
Chapter 1, Configuring VIP and Virtual Interface Redundancy	Configure VIP and virtual interface redundancy on a CSS to maintain network integrity.
Chapter 2, Configuring Adaptive Session Redundancy	Configure Adaptive Session Redundancy (ASR) on a CSS to provide stateful failover of flows.
Chapter 3, Configuring Box-to-Box Redundancy	Configure box-to-box redundancy between two mirrored CSSs.

Related Documentation

In addition to this guide, the Content Services Switch documentation includes the following publications.

Document Title	Description
Release Note for the Cisco 11500 Series Content Services Switch	This release note provides information on operating considerations, caveats, and command line interface (CLI) commands for the Cisco 11500 series CSS.
Cisco 11500 Series Content Services Switch Hardware Installation Guide	This guide provides information for installing, cabling, and powering the Cisco 11500 series CSS. In addition, this guide provides information about CSS specifications, cable pinouts, and hardware troubleshooting.
Cisco Content Services Switch Getting Started Guide	This guide describes how to perform initial administration and configuration tasks on the CSS, including: • Booting the CSS for the first time and a routine basis, and logging in to the CSS • Configuring the username and password, Ethernet management port, static IP routes, and the date and time • Configuring DNS server for hostname resolution • Configuring sticky cookies with a sticky overview and advanced load-balancing method using cookies • Installing the CSS Cisco View Device Manager (CVDM) browser-based user interface used to configure the CSS • A task list to help you find information in the CSS documentation • Troubleshooting the boot process

Document Title	Description
Cisco Content Services Switch Administration Guide	This guide describes how to perform administrative tasks on the CSS, including upgrading your CSS software and configuring the following: • Logging, including displaying log messages and interpreting sys.log messages • User profile and CSS parameters • SNMP • RMON • XML documents to configure the CSS • CSS scripting language • Offline Diagnostic Monitor (Offline DM) menu
Cisco Content Services Switch Routing and Bridging Configuration Guide	This guide describes how to perform routing and bridging configuration tasks on the CSS, including: • Management ports, interfaces, and circuits • Spanning-tree bridging • Address Resolution Protocol (ARP) • Routing Information Protocol (RIP) • Internet Protocol (IP) • Open Shortest Path First (OSPF) protocol • Cisco Discovery Protocol (CDP) • Dynamic Host Configuration Protocol (DHCP) relay agent

Document Title	Description
Cisco Content Services Switch Content Load-Balancing Configuration Guide	This guide describes how to perform CSS content load-balancing configuration tasks, including: • Flow and port mapping • Services • Service, global, and script keepalives • Source groups • Loads for services • Server/Application State Protocol (SASP) • Dynamic Feedback Protocol (DFP) • Owners • Content rules • Sticky parameters • HTTP header load balancing • Content caching • Content replication
Cisco Content Services Switch Global Server Load-Balancing Configuration Guide	This guide describes how to perform CSS global load-balancing configuration tasks, including: • Domain Name System (DNS) • DNS Sticky • Content Routing Agent • Client-Side Accelerator • Network proximity

Document Title	Description
Cisco Content Services Switch Security Configuration Guide	This guide describes how to perform CSS security configuration tasks, including: • Controlling access to the CSS • Secure Shell Daemon protocol • Radius • TACACS+ • Firewall load balancing
Cisco Content Services Switch SSL Configuration Guide	This guide describes how to perform CSS SSL configuration tasks, including: • SSL certificate and keys • SSL termination • Backend SSL • SSL initiation • HTTP data compression
Cisco Content Services Switch Command Reference	This reference provides an alphabetical list of all CLI commands including syntax, options, and related commands.

Symbols and Conventions

This guide uses the following symbols and conventions to identify different types of information.

⚠

Caution A caution means that a specific action you take could cause a loss of data or adversely impact use of the equipment.

Warning **A warning describes an action that could cause you physical harm or damage the equipment.**

Note A note provides important related information, reminders, and recommendations.

Bold text indicates a command in a paragraph.

`Courier text` indicates text that appears on a command line, including the CLI prompt.

`Courier bold text` indicates commands and text you enter in a command line.

Italics text indicates the first occurrence of a new term, book title, emphasized text, and variables for which you supply values.

1. A numbered list indicates that the order of the list items is important.
   a. An alphabetical list indicates that the order of the secondary list items is important.

- A bulleted list indicates that the order of the list topics is unimportant.
  - An indented list indicates that the order of the list subtopics is unimportant.

Obtaining Documentation

Cisco documentation and additional literature are available on Cisco.com. Cisco also provides several ways to obtain technical assistance and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation at this URL:

http://www.cisco.com/techsupport

You can access the Cisco website at this URL:

http://www.cisco.com

You can access international Cisco websites at this URL:

http://www.cisco.com/public/countries_languages.shtml

Product Documentation DVD

Cisco documentation and additional literature are available in the Product Documentation DVD package, which may have shipped with your product. The Product Documentation DVD is updated regularly and may be more current than printed documentation.

The Product Documentation DVD is a comprehensive library of technical product documentation on portable media. The DVD enables you to access multiple versions of hardware and software installation, configuration, and command guides for Cisco products and to view technical documentation in HTML. With the DVD, you have access to the same documentation that is found on the Cisco website without being connected to the Internet. Certain products also have .pdf versions of the documentation available.

The Product Documentation DVD is available as a single unit or as a subscription. Registered Cisco.com users (Cisco direct customers) can order a Product Documentation DVD (product number DOC-DOCDVD=) from Cisco Marketplace at this URL:

http://www.cisco.com/go/marketplace/

Ordering Documentation

Beginning June 30, 2005, registered Cisco.com users may order Cisco documentation at the Product Documentation Store in the Cisco Marketplace at this URL:

http://www.cisco.com/go/marketplace/

Nonregistered Cisco.com users can order technical documentation from 8:00 a.m. to 5:00 p.m. (0800 to 1700) PDT by calling 1 866 463-3487 in the United States and Canada, or elsewhere by calling 011 408 519-5055. You can also order documentation by e-mail at tech-doc-store-mkpl@external.cisco.com or by fax at 1 408 519-5001 in the United States and Canada, or elsewhere at 011 408 519-5001.

Documentation Feedback

You can rate and provide feedback about Cisco technical documents by completing the online feedback form that appears with the technical documents on Cisco.com.

You can send comments about Cisco documentation to bug-doc@cisco.com.

You can submit comments by using the response card (if present) behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Cisco Product Security Overview

Cisco provides a free online Security Vulnerability Policy portal at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

From this site, you can perform these tasks:

- Report security vulnerabilities in Cisco products.
- Obtain assistance with security incidents that involve Cisco products.
- Register to receive security information from Cisco.

A current list of security advisories and notices for Cisco products is available at this URL:

http://www.cisco.com/go/psirt

If you prefer to see advisories and notices as they are updated in real time, you can access a Product Security Incident Response Team Really Simple Syndication (PSIRT RSS) feed from this URL:

http://www.cisco.com/en/US/products/products_psirt_rss_feed.html

Reporting Security Problems in Cisco Products

Cisco is committed to delivering secure products. We test our products internally before we release them, and we strive to correct all vulnerabilities quickly. If you think that you might have identified a vulnerability in a Cisco product, contact PSIRT:

- Emergencies—security-alert@cisco.com

  An emergency is either a condition in which a system is under active attack or a condition for which a severe and urgent security vulnerability should be reported. All other conditions are considered nonemergencies.

- Nonemergencies—psirt@cisco.com

In an emergency, you can also reach PSIRT by telephone:

- 1 877 228-7302
- 1 408 525-6532

Tip We encourage you to use Pretty Good Privacy (PGP) or a compatible product to encrypt any sensitive information that you send to Cisco. PSIRT can work from encrypted information that is compatible with PGP versions 2.*x* through 8.*x*.

Never use a revoked or an expired encryption key. The correct public key to use in your correspondence with PSIRT is the one linked in the Contact Summary section of the Security Vulnerability Policy page at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

The link on this page has the current PGP key ID in use.

Obtaining Technical Assistance

Cisco Technical Support provides 24-hour-a-day award-winning technical assistance. The Cisco Technical Support & Documentation website on Cisco.com features extensive online support resources. In addition, if you have a valid Cisco service contract, Cisco Technical Assistance Center (TAC) engineers provide telephone support. If you do not have a valid Cisco service contract, contact your reseller.

Cisco Technical Support & Documentation Website

The Cisco Technical Support & Documentation website provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The website is available 24 hours a day, at this URL:

http://www.cisco.com/techsupport

Access to all tools on the Cisco Technical Support & Documentation website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register at this URL:

http://tools.cisco.com/RPF/register/register.do

Note Use the Cisco Product Identification (CPI) tool to locate your product serial number before submitting a web or phone request for service. You can access the CPI tool from the Cisco Technical Support & Documentation website by clicking the **Tools & Resources** link under Documentation & Tools. Choose **Cisco Product Identification Tool** from the Alphabetical Index drop-down list, or click the **Cisco Product Identification Tool** link under Alerts & RMAs. The CPI tool offers three search options: by product ID or model name; by tree view; or for certain products, by copying and pasting **show** command output. Search results show an illustration of your product with the serial number label location highlighted. Locate the serial number label on your product and record the information before placing a service call.

Submitting a Service Request

Using the online TAC Service Request Tool is the fastest way to open S3 and S4 service requests. (S3 and S4 service requests are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Service Request Tool provides recommended solutions. If your issue is not resolved using the recommended resources, your service request is assigned to a Cisco engineer. The TAC Service Request Tool is located at this URL:

http://www.cisco.com/techsupport/servicerequest

For S1 or S2 service requests or if you do not have Internet access, contact the Cisco TAC by telephone. (S1 or S2 service requests are those in which your production network is down or severely degraded.) Cisco engineers are assigned immediately to S1 and S2 service requests to help keep your business operations running smoothly.

To open a service request by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411 (Australia: 1 800 805 227)
EMEA: +32 2 704 55 55
USA: 1 800 553-2447

For a complete list of Cisco TAC contacts, go to this URL:

http://www.cisco.com/techsupport/contacts

Definitions of Service Request Severity

To ensure that all service requests are reported in a standard format, Cisco has established severity definitions.

Severity 1 (S1)—Your network is “down,” or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Severity 2 (S2)—Operation of an existing network is severely degraded, or significant aspects of your business operation are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Severity 3 (S3)—Operational performance of your network is impaired, but most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Severity 4 (S4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- Cisco Marketplace provides a variety of Cisco books, reference guides, documentation, and logo merchandise. Visit Cisco Marketplace, the company store, at this URL:

  http://www.cisco.com/go/marketplace/

- *Cisco Press* publishes a wide range of general networking, training and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to Cisco Press at this URL:

  http://www.ciscopress.com

- *Packet* magazine is the Cisco Systems technical user magazine for maximizing Internet and networking investments. Each quarter, Packet delivers coverage of the latest industry trends, technology breakthroughs, and Cisco products and solutions, as well as network deployment and troubleshooting tips, configuration examples, customer case studies, certification and training information, and links to scores of in-depth online resources. You can access Packet magazine at this URL:

  http://www.cisco.com/packet

- *iQ Magazine* is the quarterly publication from Cisco Systems designed to help growing companies learn how they can use technology to increase revenue, streamline their business, and expand services. The publication identifies the challenges facing these companies and the technologies to help solve them, using real-world case studies and business strategies to help readers make sound technology investment decisions. You can access iQ Magazine at this URL:

http://www.cisco.com/go/iqmagazine

or view the digital edition at this URL:

http://ciscoiq.texterity.com/ciscoiq/sample/

- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the Internet Protocol Journal at this URL:

  http://www.cisco.com/ipj

- Networking products offered by Cisco Systems, as well as customer support services, can be obtained at this URL:

  http://www.cisco.com/en/US/products/index.html

- Networking Professionals Connection is an interactive website for networking professionals to share questions, suggestions, and information about networking products and technologies with Cisco experts and other networking professionals. Join a discussion at this URL:

  http://www.cisco.com/discuss/networking

- World-class networking training is available from Cisco. You can view current offerings at this URL:

  http://www.cisco.com/en/US/learning/index.html

Configuring VIP and Virtual Interface Redundancy

This chapter describes how to plan for and configure virtual IP (VIP) redundancy and virtual interface redundancy on the CSS. Information in this chapter applies to all CSS models except where noted.

This chapter provides the following major sections:

- Overview of CSS Redundancy
- Overview of VIP and Virtual Interface Redundancy
- VIP and Virtual Interface Redundancy Configuration Quick Start
- Configuring VIP and Virtual Interface Redundancy
- Displaying VIP and Virtual Interface Redundancy Configurations

Overview of CSS Redundancy

Redundancy helps to ensure:

- High availability for your network applications
- Users do not experience long network delays or black holes due to a single point of failure.

A CSS provides three types of redundancy.

- Virtual IP (VIP) and virtual interface redundancy - Provides redundant VIP addresses and redundant virtual interfaces for fate sharing (the redundant interfaces and redundant VIPs fail over together to the backup CSS) and server default gateways. For details, see this chapter.
- Adaptive Session Redundancy (ASR) - Provides session-level redundancy (stateful failover) to continue active flows without interruption if the master CSS fails over to the backup CSS. For details, refer to Chapter 2, Configuring Adaptive Session Redundancy.
- Box-to-box redundancy - Provides chassis-level redundancy between two identically configured CSSs. For details, refer to Chapter 3, Configuring Box-to-Box Redundancy.

The following sections provide information about when (and when not) to use the different types of redundancy.

When to Use VIP and Virtual Interface Redundancy

Typically, you configure VIP redundancy on the public side of CSS peers that are positioned in front of a server farm. You configure virtual interface redundancy on the private-side interfaces attached to the Layer 2 device in front of the servers.

Configure VIP redundancy:

- With virtual interface redundancy to provide fate sharing
- When you have a common subnet between the two CSSs on which the VIPs reside
- As a prerequisite to configuring ASR (requires active-backup VIP redundancy)
- To provide active-active CSS behavior (both CSSs processing flows)

Configure interface redundancy:

- With VIP redundancy to provide fate sharing
- When you need a default gateway for the back-end servers
- Instead of VIP redundancy on the client side of the CSS when the VIPs are on a subnet different from the subnet of your uplinks

When to Use ASR

ASR provides session-level redundancy for applications where active flows (including TCP and UDP) must continue without interruption, even if the master CSS fails over to the backup CSS.

Configure ASR:

- If you require stateful failover for mission-critical applications (for example, enterprise applications; long-lived flows, such as HTTP or FTP file transfers; and e-commerce)
- After you have first configured active-backup VIP and virtual interface redundancy

When to Use Box-to-Box Redundancy

Configure box-to-box redundancy when you:

- Expect the behavior of the CSSs to be active/standby (only the master CSS processes flows)
- Can configure a dedicated Fast Ethernet (FE) link between the CSSs for the redundancy protocol

Do not configure box-to-box redundancy when you:

- Expect the behavior of the CSSs to be active-active (both CSSs processing flows). Use VIP redundancy instead.
- Cannot configure a dedicated FE link between the CSSs.

Overview of VIP and Virtual Interface Redundancy

This section provides information about:

- VIP Redundancy
- Virtual Interface Redundancy
- Fate Sharing
- Examples of VIP and Virtual Interface Redundancy Configurations

Note When using VIP or VIP interface redundancy with zone-based Global Server Load Balancing (GSLB), both the master and backup CSSs provide DNS information over their APP sessions. In this configuration, we recommend that you configure the master and backup CSSs in their own separate zones. A CSS in a GSLB configuration expects to receive zone-based information from only one CSS source within another zone. If the CSS receives information from more than one CSS within the zone, the CSS ignores the additional information, potentially causing unexpected results to DNS queries.

VIP Redundancy

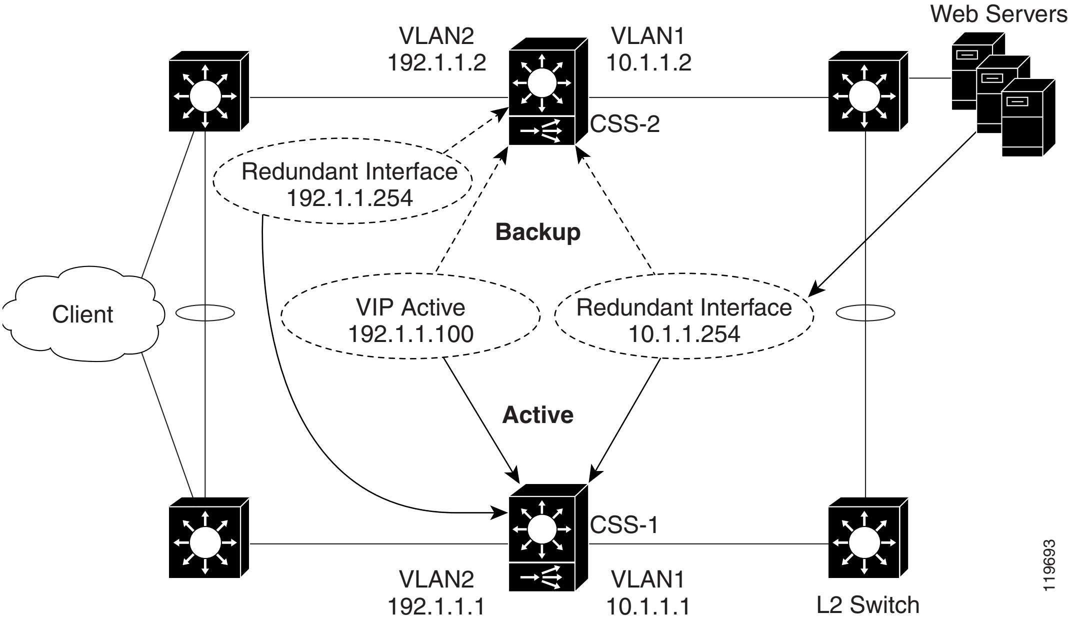
When you configure a pair of CSSs to process client requests for the same VIP address, the VIP address is considered *redundant*. A typical use of VIP redundancy is with a virtual interface redundancy configuration where the master CSS processes all client requests to a VIP with a Web-server farm behind the CSSs and connected to the CSSs through a Layer 2 switch (Figure 1-1). If the master CSS becomes unavailable, the backup CSS becomes master and processes all client requests for the VIP.

Note The CSS does not support VIP redundancy and box-to-box redundancy configurations simultaneously. For information about box-to-box redundancy, refer to Chapter 3, Configuring Box-to-Box Redundancy.

To set up CSSs for VIP redundancy, you must configure a virtual router (VR) on each CSS that will participate in the redundant configuration. A VR is an entity within a CSS with which you associate an existing VIP. A VIP becomes redundant when you associate it with a VR. You can configure a maximum of 255 VRs for each VLAN.

Note The VIP address must already exist in at least one active content rule or source group.

Figure 1-1 Example of VIP and Virtual Interface Redundancy



Virtual routers providing redundancy for a VIP address are considered a VR pair. Each VR pair has the same VR identifier (VRID) and runs on the same VLAN, but runs on a different CSS. Once the VRs are configured, the CSSs negotiate for mastership using Virtual Router Redundancy Protocol (VRRP). A VR in a redundant VIP configuration that is designated as:

- Master processes all client requests directed to the VIP

- Backup may be either a:
  - Backup VR, which forwards all client requests directed to the VIP to the master CSS
  - Shared backup VR, which processes all client requests it receives and does not forward requests for the VIP to the master CSS

A CSS designated as the master of a VIP automatically sends a gratuitous ARP for the VIP when the CSS becomes the master, either at startup or upon failover. This process enables the Layer 2 switch to learn where to forward packets that are directed to the VIP from clients. The CSS transmits one ARP request packet and one ARP reply packet for every gratuitious ARP invocation.

For an example of VIP and virtual interface redundancy, see Figure 1-1.

Virtual Interface Redundancy

Virtual interface redundancy is a form of IP address redundancy that applies only to IP interfaces (not VIPs). A typical interface IP address on a CSS defines the interface in use on a particular VLAN. In a virtual interface redundancy configuration, the CSS designated as master maintains control over the redundant virtual interface. Each CSS will also have its own circuit IP address that you can use for Telnet, SNMP, or Cisco View Device Manager (CVDM) user interface.

The typical use for virtual interface redundancy is with a VIP redundancy configuration in which:

- Web servers are positioned behind a Layer 2 switch
- CSSs with the redundant virtual interface are positioned in front of the Layer 2 switch
- The servers are configured with a default route (gateway) pointing to the redundant virtual interface IP address on the private side of the CSS
- Upstream routers use the IP address of the public-side redundant virtual interface as the next hop

You must configure VRs with the same VRIDs on the subnets that are common to both CSSs. Once you associate the new VRIDs with the redundant virtual interface IP addresses, the CSSs uses VRRP to negotiate mastership of the redundant virtual interfaces.

A CSS designated as the master of a redundant virtual interface automatically sends out gratuitous ARPs for the redundant virtual interface's IP address when the CSS becomes the master, either at startup or upon failover. This process enables a Layer 2 switch to learn where to forward packets that are directed to the redundant virtual interface from the servers or the clients and allows a server's default route to always point to the CSS designated as the master of the redundant virtual interface. The CSS transmits one ARP request packet and one ARP reply packet for every gratuitious ARP invocation.

For an example of VIP and virtual interface redundancy, see Figure 1-1.

Note Virtual interface redundancy does not support a CSS configured as a *shared* backup.

You can also configure virtual interface redundancy on the uplinks of the CSSs when the VIPs reside on a subnet different from that of the uplinks. In this case, you cannot configure VIP redundancy on the public side of the CSSs. You will need to configure static routes on the upstream routers pointing to the redundant virtual interface on the CSS as the router's next hop gateway to the subnet where the VIPs reside.

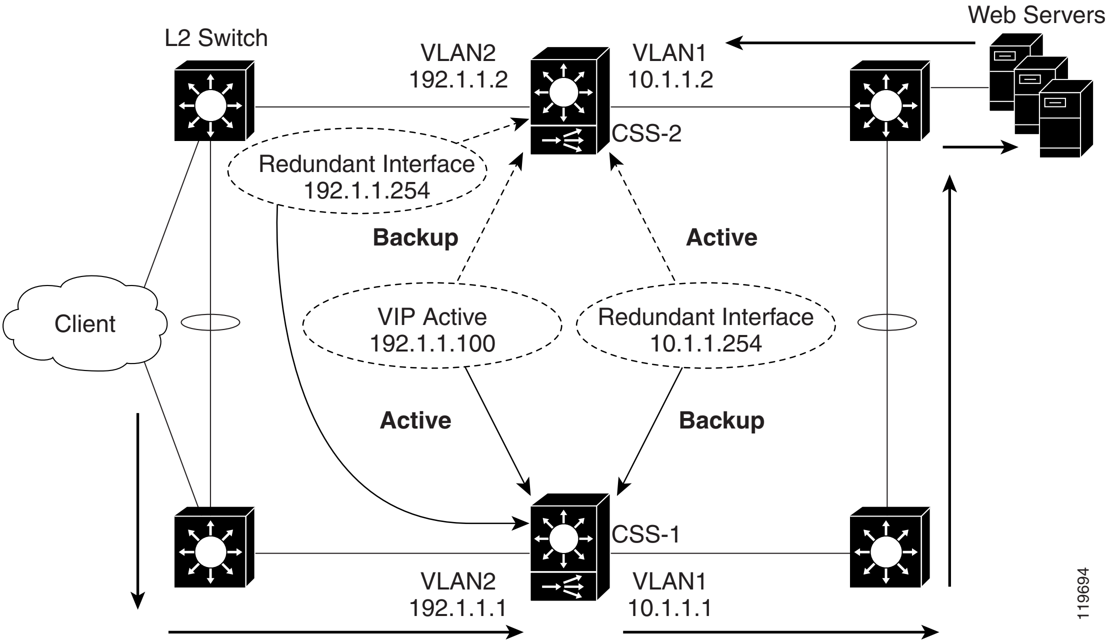
Fate Sharing

Fate sharing means that, when the redundant VIP fails over on the public side of the network from the master to the backup CSS, the redundant virtual interfaces on the public and the private sides of the network also fail over from the master to the backup CSS. If you do not configure virtual interface redundancy with VIP redundancy, asymmetric flows may result (Figure 1-2). Asymmetric flows occur when a CSS is master on the public side, but backup on the private side, which breaks the connection between the client and the server.

To ensure that the redundant VIP and the redundant virtual interfaces fail over at the same time, you must bind the front and the back instances of VRRP (the VRs) so that the same CSS processes both inbound and outbound flows. You accomplish this by defining as critical services the IP addresses of the upstream router (the CSS default gateway) and the downstream Layer 2 switch that connects to the servers.

For example, the CSS provides a scripted keepalive (ap-kal-pinglist) that will check the health of the upstream router and the downstream Layer 2 switch. When you configure this keepalive, if either device fails, the critical service goes down and the redundant VIP and the redundant virtual interfaces fail over together to the backup CSS. For details on configuring critical services, see the "Configuring a Critical Service" section.

Figure 1-2 Example of Asymmetric Flows Without Fate Sharing



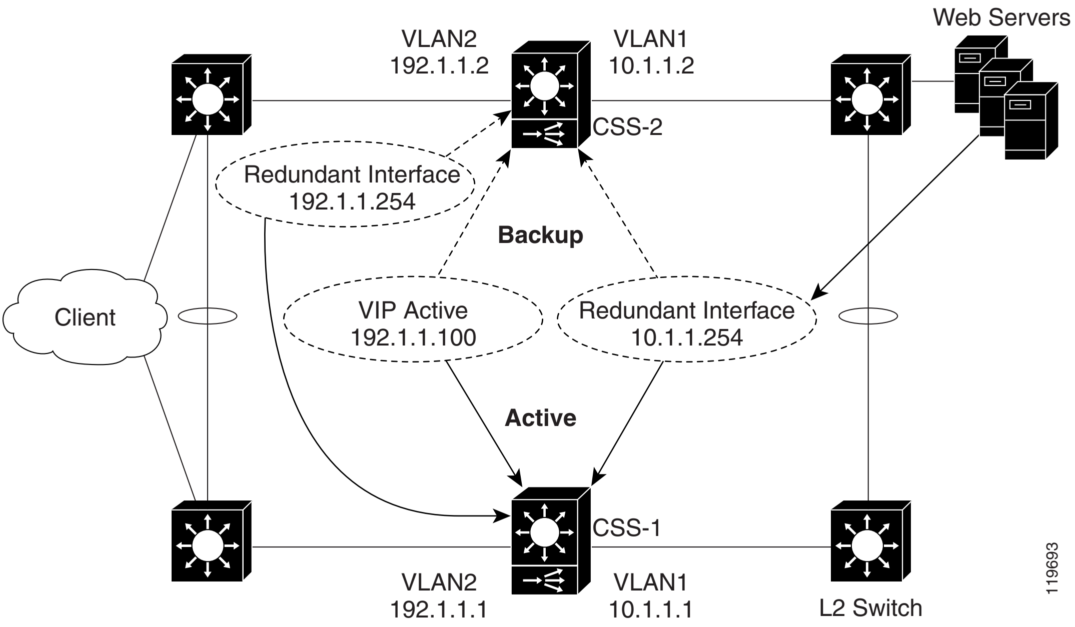
Examples of VIP and Virtual Interface Redundancy Configurations

The following sections provides examples of the most commonly used VIP and virtual interface redundancy configurations.

Active-Backup VIP and Virtual Interface Redundancy with Fate Sharing

Figure 1-3 shows an active-backup VIP and virtual interface redundancy configuration. CSS-1 is configured as the master for VIP address 192.1.1.100 and virtual interface address 10.1.1.254. If CSS-1 fails, CSS-2 (the backup CSS) will assume mastership of all flows destined to VIP address 192.1.1.100 and virtual interface address 10.1.1.254.

Figure 1-3 Example of Active-Backup VIP and Virtual Interface Redundancy



CSS-1 Configuration

```
circuit VLAN1

 ip address 10.1.1.1 255.255.255.0
  ip virtual-router 1 priority 101 preempt
  ip redundant-interface 1 10.1.1.254
  ip critical-service 1 upstream_downstream

circuit VLAN2

 ip address 192.1.1.1 255.255.255.0
  ip virtual-router 2 priority 101 preempt
  ip redundant-vip 2 192.1.1.100
  ip redundant-interface 2 192.1.1.254
  ip critical-service 2 upstream_downstream
```

CSS-2 Configuration

```
circuit VLAN1

 ip address 10.1.1.2 255.255.255.0
  ip virtual-router 1
  ip redundant-interface 1 10.1.1.254
  ip critical-service 1 upstream_downstream

circuit VLAN2

 ip address 192.1.1.2 255.255.255.0
  ip virtual-router 2
  ip redundant-vip 2 192.1.1.100
  ip redundant-interface 2 192.1.1.254
  ip critical-service 2 upstream_downstream
```

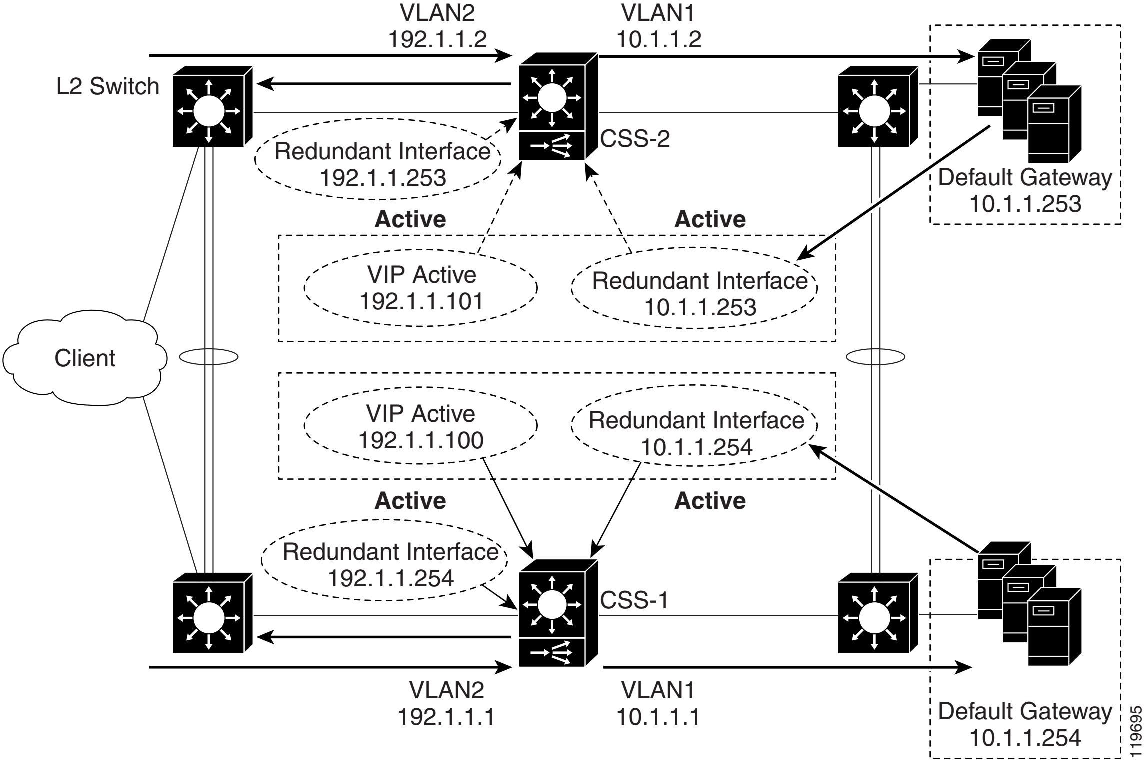
Active-Active VIP and Virtual Interface Redundancy

A CSS can serve simultaneously as a master to one VR and as a backup to a different VR. This is called active-active VIP and virtual interface redundancy. All redundant VIP addresses will share the state of the VR to which they are associated. The same VR cannot be active on both CSSs simultaneously.

Figure 1-4 shows an active-active VIP and virtual interface redundancy configuration with:

- CSS-1 configured as:
  - VLAN1 IP address 10.1.1.1.
  - VLAN2 IP address 192.1.1.1.
  - Master VR for VIP address 192.1.1.100, virtual interface address 192.1.1.254, and virtual interface address 10.1.1.254.
  - Backup VR for VIP address 192.1.1.101, virtual interface address 192.1.1.253, and virtual interface address 10.1.1.253. CSS-1 will forward all client requests it receives for VIP address 192.1.1.101 and virtual interface address 192.1.1. 253, and all server requests for virtual interface address 10.1.1.253 to CSS-2.
- CSS-2 configured as:
  - VLAN1 IP address 10.1.1.2.
  - VLAN2 IP address 192.1.1.2
  - Master VR for VIP address 192.1.1.101, virtual interface address 192.1.1.253, and virtual interface address 10.1.1.253.
  - Backup VR for VIP address 192.1.1.100, virtual interface address 192.1.1.254, and virtual interface address 10.1.1.254. CSS-2 will forward all client requests it receives for VIP address 192.1.1.100 and virtual interface address 192.1.1. 254, and all server requests for virtual interface address 10.1.1.254 to CSS-1.

Figure 1-4 Example of Active-Active VIP and Virtual Interface Redundancy



CSS-1 Configuration

```
circuit VLAN1

 ip address 10.1.1.1 255.255.255.0
  ip virtual-router 1 priority 101 preempt
  ip virtual-router 2
  ip redundant-interface 1 10.1.1.254
  ip redundant-interface 2 10.1.1.253
  ip critical-service 1 upstream_downstream
  ip critical-service 2 upstream_downstream
```

```
circuit VLAN2

 ip address 192.1.1.1 255.255.255.0
  ip virtual-router 3 priority 101 preempt
  ip virtual-router 4
  ip redundant-vip 3 192.1.1.100
  ip redundant-vip 4 192.1.1.101
  ip redundant-interface 3 192.1.1.254
  ip redundant-interface 4 192.1.1.253
  ip critical-service 3 upstream_downstream
  ip critical-service 4 upstream_downstream
```

CSS-2 Configuration

```
circuit VLAN1

 ip address 10.1.1.2 255.255.255.0
  ip virtual-router 1
  ip virtual-router 2 priority 101 preempt
  ip redundant-interface 1 10.1.1.254
  ip redundant-interface 2 10.1.1.253
  ip critical-service 1 upstream_downstream
  ip critical-service 2 upstream_downstream

circuit VLAN2

 ip address 192.1.1.2 255.255.255.0
  ip virtual-router 3
  ip virtual-router 4 priority 101 preempt
  ip redundant-vip 3 192.1.1.100
  ip redundant-vip 4 192.1.1.101
  ip redundant-interface 3 192.1.1.254
  ip redundant-interface 4 192.1.1.253
  ip critical-service 3 upstream_downstream
  ip critical-service 4 upstream_downstream
```

Shared VIP Redundancy

In a shared VIP redundancy configuration, both the master and the backup VRs process flows destined to the same VIP. However, only the master VR (CSS-1) responds to ARP requests for the VIP address 192.1.1.100.

Note Shared VIP redundancy is not supported with VRID peering. See the "Configuring VRID Peering" section.

A shared VIP redundancy configuration has the following requirements:

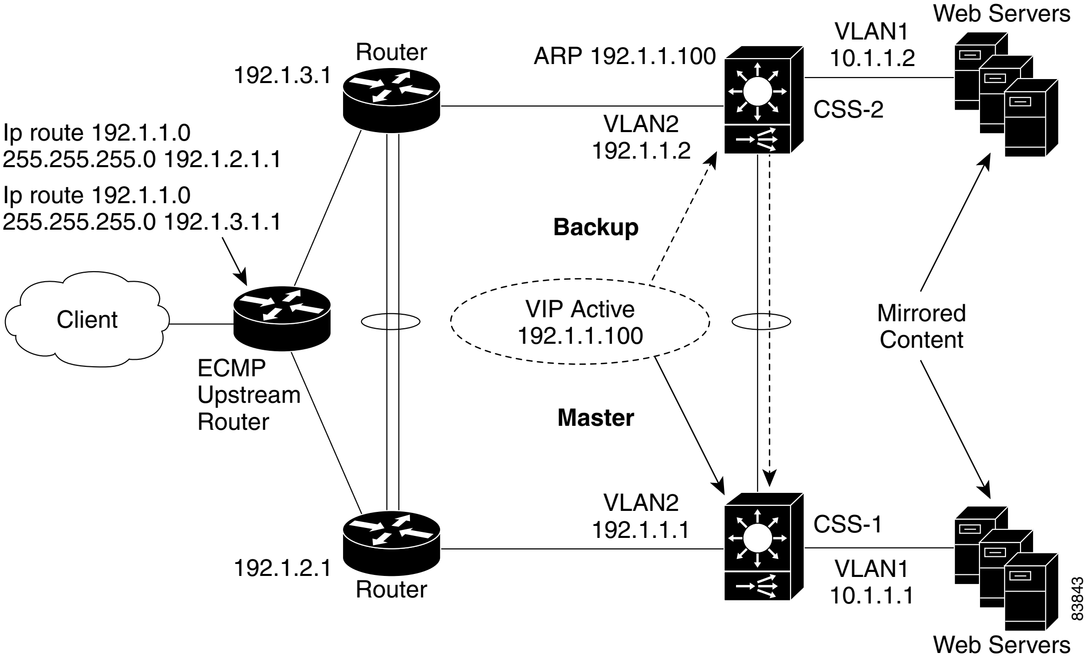
- Direct uplink connections to routers (no common Layer 2 connection between CSSs)
- Direct connection between the CSSs to enable the backup CSS to forward ARP requests to the master CSS
- Mirrored content on the servers
- Direct connection from the servers to the CSSs eliminating the need for fate sharing
- Session- or flow-based (not packet-by-packet) ECMP router upstream to preserve flow state

Figure 1-5 shows a shared VIP redundancy configuration with:

- CSS-1 configured as master VR for VIP address 192.1.1.100
- CSS-2 configured as shared backup for VIP address 192.1.1.100

Notice that CSS-2 (shared backup VR for VIP 192.1.1.100) forwards the ARP request for 192.1.1.100 to CSS-1 for a response.

Figure 1-5 Example of Shared VIP Redundancy



CSS-1 Configuration

```
circuit VLAN1

 ip address 10.1.1.1 255.255.255.0

circuit VLAN2

 ip address 192.1.1.1 255.255.255.0
  ip virtual-router 1
  ip redundant-vip 1 192.1.1.100 shared
```

CSS-2 Configuration

```
circuit VLAN1

 ip address 10.1.1.2 255.255.255.0

circuit VLAN2

 ip address 192.1.1.2 255.255.255.0
  ip virtual-router 1
  ip redundant-vip 1 192.1.1.100 shared
```

VIP and Virtual Interface Redundancy Configuration Quick Start

Table 1-1 provides a quick overview of the steps required to configure active-backup VIP and virtual interface redundancy with fate sharing for CSS-1. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI command, see the sections following Table 1-1.

Table 1-1 VIP and Virtual Interface Redundancy Configuration Quick Start

Task and Command Example

1. Enter config mode.
`# config`
`(config)#`

2. Enter service mode and configure a service to be used as a critical service or use an existing service.
`(config)# service upstream_downstream`
`(config-service[upstream_downstream])# ip address 192.1.1.75`
`(config-service[upstream_downstream])# keepalive type script ap-kal-pinglist “191.1.1.20 10.1.1.20”`
`(config-service[upstream_downstream])# keepalive frequency 2`
`(config-service[upstream_downstream])# keepalive maxfailure 2`
`(config-service[upstream_downstream])# keepalive retryperiod 2`
`(config-service[upstream_downstream])# active`

Table 1-1 VIP and Virtual Interface Redundancy Configuration Quick Start (continued)

Task and Command Example

3. Enter circuit mode for VLAN1.

```
(config)# circuit VLAN1
(config-circuit[VLAN1])#
```

4. Configure a circuit IP address.

```
(config-circuit[VLAN1])# ip address 10.1.1.1/24
(config-circuit-ip[VLAN1-10.1.1.1])#
```

5. Configure the VR. If you do not have a preference as to which router becomes master, you may leave the default priority at 100. If you have a preference, assign a higher priority to one router using the **priority** option. When you want a VR to assume mastership in all circumstances, include the **preempt** keyword.

```
(config-circuit-ip[VLAN1-10.1.1.1])# ip virtual-router 1 priority
101 preempt
```

6. Configure the redundant virtual interface.

```
(config-circuit-ip[VLAN1-10.1.1.1])# ip redundant-interface 1
10.1.1.254
```

7. Configure an existing service as a critical service for the VR.

```
(config-circuit-ip[VLAN1-10.1.1.1])# ip critical-service 1
upstream_downstream
```

8. Enter circuit mode for the next desired circuit VLAN.

```
(config)# circuit VLAN2
(config-circuit[VLAN2])#
```

9. Configure a circuit IP address.

```
(config-circuit[VLAN2])# ip address 192.1.1.1/24
(config-circuit-ip[VLAN2-192.1.1.1])#
```

10. Configure the VR.

```
(config-circuit-ip[VLAN2-192.1.1.1])# ip virtual-router 2
priority 101 preempt
```

11. Configure the redundant VIP on the VR.

```
(config-circuit-ip[VLAN2-192.1.1.1])# ip redundant-vip 2
192.1.1.100
```

Table 1-1 VIP and Virtual Interface Redundancy Configuration Quick Start (continued)

Task and Command Example
12. Configure the redundant virtual interface on the VR. `(config-circuit-ip[VLAN2-192.1.1.1])# ip redundant-interface 2 192.1.1.254`
13. Configure the critical service for the VR. `(config-circuit-ip[VLAN2-192.1.1.1])# ip critical-service 2 upstream_downstream`
14. (Recommended) Verify the configuration. `(config)# show virtual-routers`

You would configure CSS-2 in a similar manner, with the exception of the **priority** and **preempt** options of the **ip virtual-router** command.

The following running-config example shows the results of entering the commands listed in Table 1-1.

```
!************************* INTERFACE *************************
interface  2/1
  bridge vlan 2

!************************** CIRCUIT **************************
circuit VLAN1

  ip address 10.1.1.1 255.255.255.0
    ip virtual-router 1 priority 101 preempt
    ip redundant-interface 1 10.1.1.254
    ip critical-service 1 upstream_downstream

circuit VLAN2

  ip address 192.1.1.1 255.255.255.0
    ip virtual-router 2 priority 101 preempt
    ip redundant-vip 2 192.1.1.100
    ip critical-service 2 upstream_downstream
```

```
!************************* SERVICE *************************
service upstream-downstream
  ip address 192.1.1.10
  keepalive type script ap-kal-pinglist “192.1.1.20 10.1.1.20"
  keepalive frequency 2
  keepalive maxfailure 2
  keepalive retryperiod 2
  active
```

Configuring VIP and Virtual Interface Redundancy

You must configure each CSS that is part of a redundant configuration. The following sections describe how to configure VIP and virtual interface redundancy.

- Configuring a Circuit IP Interface
- Configuring a Virtual Router
- Configuring a Redundant VIP
- Configuring a Redundant Virtual Interface
- Configuring VRID Peering
- Configuring a Critical Service
- Configuring a Critical Physical Interface
- Synchronizing a VIP Redundancy Configuration

Configuring a Circuit IP Interface

Because this chapter is dedicated to configuring VIP and virtual interface redundancy, it contains only those circuit IP commands that pertain to this feature. For a complete description of all circuit IP commands, refer to the *Cisco Content Services Switch Routing and Bridging Configuration Guide*.

Before you can configure VIP and virtual interface redundancy, you must configure a circuit IP interface and assign it an IP address. To enter a specific circuit configuration mode, enter the **circuit** command and VLAN as shown in the following example:

```
(config)# circuit VLAN2
(config-circuit[VLAN2])#
```

Note When you use the **circuit** command, enter the word "VLAN" in uppercase letters and *do not* include a space between VLAN and the VLAN number (for example, VLAN1).

To assign an IP address to a circuit, use the **ip address** command from the specific circuit mode. Enter the IP address and a subnet mask in CIDR bitcount notation or dotted-decimal notation. The subnet mask range is /8 to /32. For example, to configure an IP address and subnet mask for VLAN1, enter:

```
(config-circuit[VLAN2])# ip address 192.1.1.1 /24
```

When you specify an IP address, the mode changes to the specific circuit-ip-VLAN-IP address as shown:

```
(config-circuit-ip[VLAN2-192.1.1.1])#
```

Configuring a Virtual Router

To create a virtual router (VR) on a CSS and configure the identifier and priority that is used when negotiating control of associated VIPs, use the **ip virtual-router** command. You must configure the VR before you can configure redundant VIPs.

A VR's role as a master or backup is determined during negotiations between all VRs with the same VRID and residing on the same VLAN.

Note In a VRID peering or critical phy configuration, suspending a reporter that is configured as a critical reporter causes all VRs associated with it to go down, which causes a failover from master to backup. See the "Configuring VRID Peering" and the "Configuring a Critical Physical Interface" sections.

The syntax and options for the IP interface command are:

ip virtual-router *vrid* {**priority** *number*} {**preempt**}

The variables and options are:

- *vrid* - The virtual router identifier (VRID). Enter an integer between 1 and 255. You can configure 255 VRs per VLAN. Virtual routers are considered peers when they have the same VRID and reside in the same VLAN.

- **priority** *number* - The optional priority of the VR with respect to its peers. The default priority value is 100. Enter an integer between 1 and 255. A VR with the highest priority usually becomes master. However, a higher priority VR will not assume mastership from a lower priority master unless you include the **preempt** option.

  When a VR is the master, it handles the traffic directed to its associated VIPs. To set a VR so that it will always be master, set its priority to 255 and configure it with the **preempt** option.

- **preempt** - The optional keyword that allows a higher priority VR to assert mastership over a lower priority VR. By default, a VR does not become master if the current master has a lower priority.

  For example, if a CSS with a VR that has a low priority boots before other CSSs, that VR becomes the master. When another CSS with a VR that has a higher priority boots, it will not take the mastership from the first router unless you specify the **preempt** option on the higher priority VR. This option does not have an effect if the priority of the two VRs is identical. You can use this option with or without the **priority** option. You can configure only one VR as the master of a particular VIP.

Caution Never configure the **preempt** option on the same VR on both CSSs. Such a configuration may result in both CSSs becoming master, which will cause network problems.

Because a VR's priority is dependent on the state of the critical services, the priority field status in the **show virtual router** display may be different than the priority you configured. The priority may be different when you:

- Assign a priority of 255 to a VR and that VR gains mastership, the CSS automatically reconfigures that VR's priority to 254. This action ensures that you can assign a different VR a priority of 255.
- Configure critical services. The critical service types are:
  - **scripted -** the priority changes to 0 when one service in the scripted group goes down.
  - **redundancy uplink** - the priority changes to 0 when all of the services in the uplink group go down.
  - **local** - the priority changes to 0 when all of the services in the local group go down. Local services include all services other than scripted and uplink.

For information about configuring critical services, see the "Configuring a Critical Service" section.

For example:

```
(config-circuit-ip[VLAN2-192.1.1.1])# ip virtual-router 2 priority 1
preempt
```

To remove the VR from the CSS, enter:

```
(config-circuit-ip[VLAN2-192.1.1.1])# no ip virtual-router 2
```

Configuring a Redundant VIP

To associate an existing VIP with a VR and, if required, configure the VR as a shared backup, use the **ip redundant-vip** command. A shared backup VR processes client requests. A redundant VIP configuration can consist of only two CSSs.

Note Before you use this command, the VIP must already be configured in at least one active content rule or source group. Additionally, if you defined the content rule or source group VIP using the range option, you must configure an identical range for the redundant VIP. For information about configuring VIPs in content rules and source groups, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

The syntax of this IP mode command is:

ip redundant-vip *vrid vip_address* {**range** *number*} {**shared**}

The variables and options are:

- *vrid* - The ID for an existing VR.
- *vip_address* - The address for the redundant VIP. This address must already be configured in at least one active content rule or source group. Enter an IP address in dotted-decimal notation (for example, 192.1.1.100).
- **range** *number* - The optional keyword and variable if an IP address range is specified in the content rule or source group. You cannot specify a range that differs from the range in the content rule. Also, you cannot specify address ranges that overlap. Enter a number from 0 to 65535. The default is 1.

- **shared** - The optional keyword to enable shared VIP redundancy. When you use this option, the master and backup VRs share the processing of traffic directed to the VIP, so the backup does not forward packets to the master. Configure each VIP identically on both CSSs.

Caution Do not connect Layer 2 devices between the CSSs and the routers in a shared VIP redundancy configuration. In addition, each router must be connected to only one CSS. Otherwise, all traffic will go to the master CSS, thus defeating the purpose of shared VIP redundancy.

For example:

```
(config-circuit-ip[VLAN2-192.1.1.1])# ip redundant-vip 2 192.1.1.100
range 10 shared
```

To remove a VIP from a VR, enter:

```
(config-circuit-ip[VLAN1-192.1.1.1])# no ip redundant-vip 1
192.1.1.100
```

Configuring a Redundant Virtual Interface

Servers use the IP address of a redundant virtual interface as a default gateway to guarantee that packets are sent to the CSS containing the master VR. To accomplish this goal, configure a redundant virtual interface with the same VR as a redundant VIP that is configured in a rule that refers to the server. This configuration ensures that the master CSS for a VIP is the same CSS that is master for the redundant virtual interface. If the master CSS fails over to the backup CSS, both the VIP and the redundant interface fail over together. This configuration is called *fate sharing*. For details, see the "Fate Sharing" section.

We recommend that you also configure a redundant virtual interface on the public side of each CSS. This configuration provides a single IP address as the next hop for the upstream routers.

To configure a redundant virtual interface, use the **ip redundant-interface** command.

The syntax of this IP mode command is:

ip redundant-interface *vrid ip_address*

The variables are:

- *vrid* - Identifier of a previously-configured VR.
- *ip_address* - Address for the redundant interface. Enter an IP address in dotted-decimal notation (for example, 10.1.1.254).

Note You cannot use an IP address that already exists for a VIP, redundant VIP, source group, service, log host, or IP interface address on a circuit. If you do, the following error message appears: `Address conflicts with local I/F, VIP, service, or source group.`

For example:

```
(config-circuit-ip[VLAN1-10.1.1.1])# ip redundant-interface 1
10.1.1.254
```

To remove an interface from a VR, enter:

```
(config-circuit-ip[VLAN1-10.1.1.1])# no ip redundant-interface 1
10.1.1.254
```

Note The CSS does not support a traceroute of a redundant IP interface.

Configuring VRID Peering

To ensure that the state of the virtual routers (VRs) on both the public and the private sides of a CSS remain synchronized, configure VRID peering in a VIP and virtual interface redundancy configuration. VRID peering groups VRs that are configured on the same CSS so that, when one VR in the group changes state (for example, from master to backup), all VRs in the group change state at the same time. This feature helps to prevent asymmetric flows, which cause network address translation (NAT) to fail and break the client-server connection.

This section contains the following topics:

- Background
- Overview of VRID Peering
- Configuration Requirements and Restrictions

- VRID Peering Quick Start
- Configuring a Reporter
- Configuring the Reporter Type
- Configuring the Virtual Routers That You Want to Monitor
- Activating a Reporter
- Suspending a Reporter
- Configuring a Critical Reporter
- Resetting the Reporter State Transitions Counter

Background

In a VIP and virtual interface redundancy configuration, you typically configure a pair of virtual routers (VRs) running VRRP to negotiate mastership of the redundant VIPs on the client side of the network and another pair of VRs to negotiate mastership of the redundant interfaces on the server side of the network. In addition, you can use critical services to provide fate sharing so that the redundant VIPs and the redundant interfaces change state together from master to backup. A CSS uses polling keepalives to monitor the states of critical services.

Because a link on either the public side or the private side of a CSS can go down and then up very quickly in between keepalive polling times, it is possible for a VR on one CSS to be the master for the redundant VIP and a VR on the other CSS to be the master for the redundant virtual interface. This loss of synchronization between VRs can occur even when you have configured critical services. While this unsynchronized VR state is permitted, it may not be desirable in all cases.

Unsynchronized VRs produce asymmetric flows, which cause NAT to fail. In this case, packets from a server will be routed to the CSS that is master for the redundant virtual interface (default gateway), while packets from a client will be routed to the other CSS that is master for the redundant VIP.

Overview of VRID Peering

VRID peering works by grouping two or more VR peers with a software agent called a *reporter* that monitors the states of the VRs. VRs are considered peers when they are configured on the same CSS. The reporter ensures that the states of the VR peers are synchronized by sending internal state update messages to the monitored VRs when necessary.

The internal state of a virtual router is called an independent VR state because it does not depend on the state of its VR peers. The VR peer reporter state is called a dependent VR state because it depends on the states of all VRs configured on the reporter.

Table 1-2 lists the independent VR states and describes the conditions required for each state.

Table 1-2 Independent VR States and Conditions

Independent State	Conditions
Down	Circuit where the VR runs is down or one of the critical services configured on the VR failed
Backup	Circuit where the VR runs is up, there are no failures on the critical services configured on the VR, and VRRP has determined that the VR should be in the backup state
Master	Circuit where the VR runs is up, there are no failures on the critical services configured on the VR, and VRRP has determined that the VR should be in the master state

The reporter that is responsible for monitoring the VR peers determines the dependent state of the VRs. Table 1-3 lists the dependent VR states and describes the conditions required for each state.

Table 1-3 Dependent VR States and Conditions

Dependent State	Conditions
Down	At least one of the VR peers in a reporter group is in the Down state. The VRs will hold the VIPs and redundant virtual interfaces that they control in the Down state.
Backup	At least one of the VR peers in a reporter group is in the Backup state and none is in the Down state. The VRs will hold the VIPs and redundant virtual interfaces that they control in the Backup state.
Master	All VR peers in a reporter group are in the Master state. The VRs transmit VRRP messages announcing that they are in the Master state. VIPS and redundant virtual interfaces controlled by the VRs become masters.

When two or more VRs are configured on a reporter, their dependent state is that of the lowest independent state of all the VR peers in the group. Table 1-4 shows the effects of independent VR states on reporter dependent states for two VRs.

Table 1-4 Effect of Independent VR States on Reporter Dependent States

VR1 Independent State	VR2 Independent State	Reporter Dependent State
Down	Down	Down
Down	Backup	Down
Down	Master	Down
Backup	Down	Down
Backup	Backup	Backup
Backup	Master	Backup
Master	Down	Down
Master	Backup	Backup
Master	Master	Master

Configuration Requirements and Restrictions

The following requirements and recommendations apply to the configuration and use of VRID peering on a Cisco 11500 series CSS.

- Ensure that you have configured VIP and virtual interface redundancy properly. See the "Configuring VIP and Virtual Interface Redundancy" section.
- VRID peering is not supported with shared VIP redundancy.
- Ensure that a VR exists before you attempt to configure it on a reporter. See the "Configuring a Virtual Router" section.
- All VRs associated with a VRID peering reporter must have the same priority and preempt configurations.
- Do not configure the same IP address and VRID on more than one reporter.
- You can configure a maximum of 128 reporters on a CSS.
- You can configure a maximum of four reporters of type VRID peer on a CSS.
- You can configure a maximum of eight VRIDs on a reporter of type VRID peer.

VRID Peering Quick Start

Table 1-5 provides a quick overview of the steps required to configure VRID peering. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI command, see the sections following Table 1-5.

Table 1-5 VRID Peering Configuration Quick Start

Task and Command Example

1. Enter config mode.
`# config`
`(config)#`

2. Enter reporter configuration mode and create a new reporter. See the "Configuring a Reporter" section.
`(config)# reporter r1`
`(config-reporter[r1])#`

3. Configure the VRID peering type on the reporter. See the "Configuring the Reporter Type" section. `(config-reporter[r1])# type vrid-peering`

4. Specify the VRs that you want to monitor. See the "Configuring the Virtual Routers That You Want to Monitor" section.
`(config-reporter[r1])# vrid 192.168.100.5 1`
`(config-reporter[r1])# vrid 172.16.27.12 2`

5. Activate the reporter. See the "Activating a Reporter" section. `(config-reporter[r1])# active`

6. Associate the reporter with an existing VRID. See the "Configuring a Critical Reporter" section.
`(config-circuit-ip[VLAN1-192.168.100.5])# ip critical-reporter 1 r1`
`(config-circuit-ip[VLAN2-172.16.27.12])# ip critical-reporter 2 r1`

7. (Recommended) Verify the reporter configuration. See the "Displaying a Reporter Configuration in the Running-Config" section. `(config-circuit-ip[VLAN2-172.16.27.12])# show reporter r1`

Table 1-5 VRID Peering Configuration Quick Start (continued)

Task and Command Example
8. (Recommended) Verify the VRID peering configuration. See the "Displaying Critical Reporter Information" section. `(config-circuit-ip[VLAN2-172.16.27.12])# show running-config reporter`

The following running-config example shows the results of entering the commands in Table 1-5.

```
!************************* CIRCUIT *************************
circuit VLAN1

  ip address 192.168.100.5 255.255.255.0
    ip virtual-router 1 priority 101 preempt
    ip critical-reporter 1 r1

circuit VLAN2

  ip address 172.16.27.12 255.255.255.0
    ip virtual-router 2 priority 101 preempt
    ip critical-reporter 2 r1

!************************* REPORTER *************************
reporter r1
  type vrid-peering
  vrid 192.168.100.5 1
  vrid 172.16.27.12 2
  active
```

Configuring a Reporter

A reporter is a software agent that monitors the states of all VRs associated with it. When a VR state change is necessary, the reporter sends a state update message to its associated VRs. To configure a reporter and enter reporter configuration mode, use the **reporter** command in global configuration mode. You can configure a maximum of 128 reporters on a CSS.

This command has the following syntax:

reporter *reporter_name*

The *reporter_name* variable specifies the name of the reporter you are creating. Enter an unquoted text string with no spaces from 1 to 31 characters.

For example, enter:

```
(config)# reporter r1
```

To remove an existing reporter from the running-config, enter:

```
(config-reporter[r1])# no reporter r1
```

Note If you remove a reporter from the running-config using the **no reporter** command, the CSS removes all the attributes associated with that reporter from the running-config.

Configuring the Reporter Type

A VRID peer is a type of reporter that monitors the states of associated VRs and ensures that the VR states are synchronized. To configure the reporter type, use the **type** command in reporter configuration mode. You can configure a maximum of four reporters of type **vrid-peering** on a CSS.

Note The CSS supports reporters of type **vrid-peering** only with non-shared active-backup and active-active VIP redundancy configurations.

This command has the following syntax.

type *reporter_type*

For example, to configure a reporter as a VRID peering type, enter:

```
(config-reporter[r1])# type vrid-peering
```

To remove the VRID peering type or to reconfigure the reporter type as a critical phy type:

1. Suspend the reporter. See the "Suspending a Reporter" section.

   Note If the reporter is configured as a critical reporter, suspending it causes the associated VR to transition to Backup or Down.

2. Remove the VRID peering reporter attributes using the **no vrid** *ip_address vrid* command. See the "Configuring the Virtual Routers That You Want to Monitor" section.
3. Remove the VRID peering reporter type using the **no type** command or reconfigure the reporter type as a critical phy type using the **type critical-phy-all-up** or the **type critical-phy-any-up** command. For details about configuring a critical phy, see the "Configuring a Critical Physical Interface" section.
4. If you reconfigured the reporter type as a critical phy type, add the physical interfaces using the **phy** *interface_name* command, then activate the reporter using the **active** command. See the "Configuring the Physical Interfaces That You Want to Monitor" section and the "Activating a Reporter" section.

Configuring the Virtual Routers That You Want to Monitor

To configure the VRs that you want a reporter to monitor, use the **vrid** command in reporter configuration mode. This command allows you to configure a maximum of eight VRIDs on a reporter of type **vrid-peering**.

This command has the following syntax:

vrid *ip_address vrid*

Note You cannot configure the same circuit IP address and VRID on more than one reporter.

The variables for this command are:

- *ip_address* - The circuit IP address of the CSS in dotted-decimal notation.
- *vrid* - The VR identifier (VRID). Enter the VRID of an existing VR. You can configure a maximum of eight VRIDs on one reporter. For details on configuring a VR, see the "Configuring a Virtual Router" section.

For example:

```
(config-reporter[r1])# vrid 192.168.12.7 1
```

Note You cannot remove the last remaining VR from an active reporter. To remove the VR, first suspend the reporter, and then remove the VR.

To remove a VR from a reporter, enter:

```
(config-reporter[r1])# no vrid 192.168.12.7 1
```

Activating a Reporter

You must activate a reporter before it can monitor the state of its configured VRIDs. To activate a reporter, use the **active** command in reporter configuration mode. A reporter remains in the Suspended state until you activate it.

For example, enter:

```
(config-reporter[r1])# active
```

Suspending a Reporter

You can suspend an active reporter to temporarily stop using the reporter or to change the reporter configuration. To suspend the reporter, enter the **suspend** command in reporter configuration mode. When you are ready to resume using the reporter again, reactivate the reporter using the **active** command. See the "Activating a Reporter" section.

Note Suspending a reporter that is configured as a critical reporter causes all VRs associated with it to go down, which causes a failover from master to backup. See the "Configuring a Critical Reporter" section.

For example, enter:

```
(config-reporter[r1])# suspend
```

Configuring a Critical Reporter

To associate a reporter with a VR, use the **ip critical-reporter** command in circuit configuration mode. You can associate more than one critical reporter with a VR provided that the critical reporters are of different types. See the "Configuring a Critical Physical Interface" section.

If any critical reporter is suspended or goes down, all VRs associated with it go down. To ensure that the VR states are synchronized, configure a critical reporter on both the front-side and the back-side VRs.

The syntax of this command is:

ip critical-reporter *vrid reporter_name*

The variables are:

- *vrid* - The virtual router identifier (VRID) of an existing VR. Enter an integer between 1 and 255.
- *reporter_name* - The name of an existing reporter. Enter an unquoted text string with no spaces from 1 to 31 characters.

For example, to associate reporter r1 with a VR that has a VRID of 1, enter:

```
(config-circuit-ip[VLAN1-192.168.7.9])# ip critical-reporter 1 r1
```

To remove a critical reporter from the running-config, enter:

```
(config-circuit-ip[VLAN1-192.168.7.9])# no ip critical-reporter 1 r1
```

Configuring a Critical Service

Configure a critical service to monitor the health of upstream and downstream devices. When one or all critical services go down (depending on the type of critical service you configure), the associated VR also goes down, which causes a failover from master CSS to backup CSS. There are three types of critical services that you can configure:

- A scripted critical service, as defined by the **(config-service) keepalive type script** command or the **(config-service) keepalive type named** command, that is constantly scanning for service and network availability. The keepalive sets the service to a down state whenever network or service availability is a problem. The VR goes down if *any* associated scripted service goes down.

  The CSS provides a scripted keepalive called **ap-kal-pinglist** that you can use to check the health of, for example, an upstream router (192.1.1.254) running Hot Standby Router Protocol (HSRP) and a downstream Layer 2 switch (10.1.1.200) as critical services.

  For example, create the service as follows:

  ```
  (config)# service upstream_downstream
  (config-service[upstream_downstream])# ip address 192.1.1.254
  (config-service[upstream_downstream])# keepalive type script
  ap-kal-pinglist "192.1.1.254 10.1.1.200"
  (config-service[upstream_downstream])# active
  ```

- A redundancy uplink critical service, as defined by the **(config-service) type redundancy-up** command. The VR goes down when *all* associated redundancy uplink services go down regardless of any configured keepalive type. Refer to Chapter 3, Configuring Box-to-Box Redundancy, in the "Configuring Multiple Redundant Uplink Services" section.

  Note You cannot add redundant uplink services to a content rule.

- Local critical services for any service other than scripted or redundancy uplink, such as a Web service. The VR goes down when *all* associated local critical services go down.

To associate a critical service with a VR, use the **ip critical-service** command. The syntax of the **ip critical-service** command is:

ip critical-service *vrid service_name*

The variables are:

- *vrid* - The ID for an existing VR.
- *service_name* - The name of the service. To see a list of services, enter **ip critical-service** *vrid* **?**.

For example:

```
(config-circuit-ip[VLAN2-192.1.1.1])# ip critical-service 1
upstream_downstream
```

To remove a critical service from a VR, enter:

```
(config-circuit-ip[VLAN2-192.1.1.1])# no ip critical-service 1
upstream_downstream
```

Note If you configure different critical services on the two CSSs and you intend to synchronize the CSS configurations using the commit_VipRedundConfig script, do not use the **-a** script argument. This argument copies the master CSS configuration to the backup CSS configuration, which makes the two configs identical. For details on synchronizing a VIP and virtual interface redundancy configuration, see the "Synchronizing a VIP Redundancy Configuration" section.

The **show service** command displays the current service type only. It does, however, display the keepalive type, so you can determine from it the behavior of a configured critical service. To display critical service-specific information, use the **show critical-services** command. See the "Displaying IP Critical Services" section.

SNMP values returned for services show the current service type only. To determine the critical service behavior of a particular service, you need to examine the service keepalive type. For more information about SNMP, refer to the *Cisco Content Services Switch Administration Guide*.

Configuring a Critical Physical Interface

Configure a critical physical interface (critical phy) on a CSS to provide an additional failover catalyst for a virtual router (VR) in a VIP and interface redundancy configuration. A critical phy improves the failover time of a VR (as compared with a critical service) by reacting quickly to a Down state of monitored physical interfaces. This feature is intended to complement critical services, not replace them. For details on critical services, see the "Configuring a Critical Service" section.

This section contains the following topics:

- Overview
- Configuration Requirements and Restrictions
- Critical Phy Quick Start
- Configuring a Reporter
- Configuring the Reporter Type
- Configuring the Physical Interfaces That You Want to Monitor
- Activating a Reporter
- Suspending a Reporter
- Configuring a Critical Reporter

Overview

A critical phy monitors the health of its associated physical interfaces and causes a VR to fail over to the backup CSS if one or all (depending on the configuration) monitored interfaces go down. Unlike a critical service, a critical phy does not depend on the state of a keepalive for its operation. Therefore, a critical phy is not susceptible to reporting delays and packet loss due to network congestion and packet storms, which, in the case of a critical service, may cause the CSS to incorrectly report a server as unavailable. When you associate a critical phy with a VR, the critical phy provides rapid VR failover in the event of a physical link failure.

To configure a critical phy, you create a software agent called a *reporter*, a general-purpose monitoring mechanism. Then you specify the reporter type of **critical-phy** and configure the reporter to monitor the health of one or more physical interfaces. To complete the configuration, you modify your VIP and virtual interface redundancy configuration to associate a critical reporter with an existing VR.

Configuration Requirements and Restrictions

The following requirements and recommendations apply to the configuration and use of critical phys on a Cisco 11500 series CSS:

- Ensure that VIP and virtual interface redundancy is configured properly (see the "Configuring VIP and Virtual Interface Redundancy" section).
- If you associate more than one critical reporter with the same VR, ensure that you do not configure the same physical interfaces (ports) on two different reporter types (for example, ports 1/1 and 1/2 on a reporter of type **critical-phy-all-up** and ports 1/1 and 1/2 on a reporter of type **critical-phy-any-up**). Otherwise, unexpected VR failovers may occur.
- Do not configure the Ethernet management port or the console port as critical-phy interfaces to be monitored by a reporter.
- Do not use critical-phy interfaces as InterSwitch Communications (ISC) ports in an ASR environment.
- Do not use critical-phy interfaces as ports to be monitored by the Switch Port Analyzer (SPAN) feature.
- Do not use critical-phy interfaces as redundancy-phy interfaces in a box-to-box redundancy configuration.

Critical Phy Quick Start

Table 1-6 provides a quick overview of the steps required to configure a critical phy on a reporter. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI command, see the sections following Table 1-6.

Table 1-6 Critical Phy Configuration Quick Start

Task and Command Example

1. Enter config mode.
`# config`
`(config)#`

2. Enter reporter configuration mode and create a new reporter. See the "Configuring a Reporter" section.
`(config)# reporter r1`
`(config-reporter[r1])#`

3. Configure the reporter type. See the "Configuring the Reporter Type" section. `(config-reporter[r1])# type critical-phy-all-up`

4. Specify the physical interfaces that you want to monitor.
`(config-reporter[r1])# phy 1/1`
`(config-reporter[r1])# phy 1/2`

5. Activate the reporter. `(config-reporter[r1])# active`

6. Exit reporter mode.
`(config-reporter[r1])# exit`
`(config)#`

7. Enter circuit configuration mode.
`(config)# circuit VLAN1`
`(config-circuit[VLAN1])# ip address 192.168.7.9`
`(config-circuit-ip[VLAN1-192.168.7.9])#`

8. Associate the reporter with an existing VRID. `(config-circuit-ip[VLAN1-192.168.7.9])# ip critical-reporter 1 r1`

Table 1-6 Critical Phy Configuration Quick Start (continued)

Task and Command Example
9. (Recommended) Verify the reporter configuration. `(config-circuit-ip[VLAN1-192.168.7.9])#` **`show reporter r1`**
10. (Recommended Verify the critical phy configuration. `(config-circuit-ip[VLAN1-192.168.7.9])#` **`show running-config reporter`**
11. (Recommended) Verify the VR and critical reporter configurations. `(config-circuit-ip[VLAN1-192.168.7.9])#` **`show running-config circuit`**

The following running-config example shows the results of entering the commands in Table 1-6.

```
!************************* CIRCUIT *************************
circuit VLAN1

  ip address 192.168.7.9 255.255.255.0
    ip virtual-router 1 priority 101 preempt
    ip critical-reporter 1 r1

!************************* REPORTER *************************
reporter r1
  type vrid-peering
  phy 1/1
  phy 1/2
  active
```

Configuring a Reporter

A reporter is a software agent that the CSS uses to monitor the health of physical interfaces when you configure the reporter as a critical phy type and associate physical interfaces with it. To configure a reporter and enter reporter configuration mode, use the **reporter** command in global configuration mode. This command has the following syntax:

reporter *reporter_name*

The *reporter_name* variable specifies the name of the reporter you are creating. Enter an unquoted text string with no spaces from 1 to 31 characters.

For example, enter:

```
(config)# reporter r1
```

To remove an existing reporter and all of its attributes from the running-config, enter:

```
(config-reporter[r1])# no reporter r1
```

Configuring the Reporter Type

A critical phy is a type of reporter that determines how the reporter reacts to a Down state of the associated physical interfaces. To configure a critical phy type on a reporter, use the **type** command in reporter configuration mode. This command has the following syntax.

type *reporter_type*

Note If you associate more than one reporter with the same VR, we recommend that you do not configure the same physical interfaces (ports) on two different reporter types (for example, ports 1/1 and 1/2 on a reporter of type **critical-phy-all-up** and ports 1/1 and 1/2 on a reporter of type **critical-phy-any-up**). Otherwise, unexpected VR failovers may occur.

The *reporter_type* variable has one of the following values:

- **critical-phy-all-up** - If any critical interface goes down, the reporter goes down and mastership of the associated VR transitions from the master CSS to the backup CSS. To prevent a VR failover, all interfaces must remain up.
- **critical-phy-any-up** - If all associated critical interfaces go down, the reporter goes down and mastership of the associated VR transitions from the master CSS to the backup CSS. As long as one critical interface stays up, the reporter and the VR remain up.

You can change the critical phy reporter type without suspending the reporter. However, if you want to reconfigure a critical phy reporter as a VRID peering reporter, you must first suspend the reporter to remove the critical phy reporter attributes. See the "Suspending a Reporter" section.Then you can configure the reporter as a VRID peering reporter and activate it. For details about VRID peering, see the "Configuring VRID Peering" section.

You can configure a maximum of 128 reporters of any combination of types on a CSS, depending on available memory.

When you configure a critical phy, the states of the monitored physical interfaces affect the reporter state, which in turn affects the VR state, depending upon the type of configured reporter as shown in Table 1-7.

Table 1-7 Effect of Interface State on Reporter and Virtual Router State Based on Reporter Type

Reporter Type	Interface State	Reporter State	Virtual Router State
critical-phy-all-up	All Up	Up	Up
critical-phy-all-up	All Down	Down	Down
critical-phy-all-up	One or More Down	Down	Down
critical-phy-any-up	All Up	Up	Up
critical-phy-any-up	All Down	Down	Down
critical-phy-any-up	One or More Up	Up	Up

For example, enter:

```
(config-reporter[r1])# type critical-phy-all-up
```

To remove the critical phy type (either **critical-phy-all-up** or **critical-phy-any-up**) or to reconfigure the reporter type as a **vrid-peering** type:

1. Suspend the reporter. See the "Suspending a Reporter" section.

   Note If the reporter is configured as a critical reporter, suspending it causes the associated VR to transition to Backup or Down.

2. Remove the critical phy reporter attributes using the **no phy** *interface_name* command. See the "Configuring the Physical Interfaces That You Want to Monitor" section.
3. Remove the **critical phy** reporter type using the **no type** command or reconfigure the reporter type as a VRID peering type using the **type vrid-peering** command. For details about configuring VRID peering, see the "Configuring VRID Peering" section.

4. If you reconfigured the reporter type as a VRID peering type, add the VRIDs using the **vrid** *ip_address vrid* command, and then activate the reporter using the **active** command. See the "Configuring the Virtual Routers That You Want to Monitor" section and the "Activating a Reporter" section.

Configuring the Physical Interfaces That You Want to Monitor

To configure one or more physical interfaces that you want a reporter to monitor, use the **phy** command in reporter configuration mode. You can configure a maximum of 128 interfaces on a reporter.

Note If you associate more than one reporter with the same VR, we recommend that you do not configure the same physical interfaces (ports) on two different reporter types (for example, ports 1/1 and 1/2 on a reporter of type **critical-phy-all-up** and ports 1/1 and 1/2 on a reporter of type **critical-phy-any-up**). Otherwise, unexpected VR failovers may occur.

This command has the following syntax:

phy *interface_name*

The *interface_name* variable is the name of the physical interface that you want to monitor. Enter an interface name in interface port format (for example, e1 on a CSS 11501) or slot/port format (for example, 1/1 on a CSS 11503 and CSS 11506). See the following configuration examples.

To configure Ethernet port 1 on a CSS 11501, enter:

```
(config-reporter[r1])# phy e1
```

To configure Ethernet port 1 on a CSS 11503 or 11506, enter:

```
(config-reporter[r1])# phy 1/1
```

Note You cannot remove the last remaining physical interface from an active reporter. To remove the interface, first suspend the reporter, and then remove the interface.

To remove Ethernet port 1 from the list of interfaces to monitor on a CSS 11501, enter:

```
(config-reporter[r1])# no phy e1
```

To remove Ethernet port 1 from the list of interfaces to monitor on a CSS 11503 or 11506, enter:

```
(config-reporter[r1])# no phy 1/1
```

Activating a Reporter

Before a CSS can use a reporter to monitor the health of the configured critical interfaces, you must activate the reporter using the **active** command. A reporter remains in a suspended state until you activate it.

For example, enter:

```
(config-reporter[r1])# active
```

Suspending a Reporter

You can suspend an active reporter to temporarily stop using the reporter or to change the reporter configuration. To suspend the reporter, enter the **suspend** command in reporter configuration mode. When you are ready to resume using the reporter again, reactivate the reporter using the **active** command. See the "Activating a Reporter" section.

For example, enter:

```
(config-reporter[r1])# suspend
```

Note Suspending a reporter that is configured as a critical reporter causes all VRs associated with it to go down, which causes a failover from master to backup. See the "Configuring a Critical Reporter" section.

Configuring a Critical Reporter

To associate a reporter with a VR, use the **ip critical-reporter** command in circuit configuration mode. You can associate more than one critical reporter with a VR. If any critical reporter is suspended or goes down, all VRs associated with it go down and cause a failover from master to backup.

Note If you associate more than one reporter with the same VR, we recommend that you do not configure the same physical interfaces (ports) on two different reporter types (for example, ports 1/1 and 1/2 on a reporter of type **critical-phy-all-up** and ports 1/1 and 1/2 on a reporter of type **critical-phy-any-up**). Otherwise, unexpected VR failovers may occur.

The syntax of this command is:

ip critical-reporter *vrid reporter_name*

The variables for this command are:

- *vrid* - The virtual router identifier (VRID) of an existing VR. Enter an integer between 1 and 255. Virtual routers are considered peers when they have the same VRID and reside in the same VLAN.
- *reporter_name* - The name of an existing reporter. Enter an unquoted text string with no spaces from 1 to 31 characters.

For example, enter:

```
(config-circuit-ip[VLAN1-192.168.7.9])# ip critical-reporter 1 r1
```

To remove a critical reporter, enter:

```
(config-circuit-ip[VLAN1-192.168.7.9])# no ip critical-reporter 1 r1
```

Synchronizing a VIP Redundancy Configuration

To ensure that your remote CSS can perform the same tasks as your local CSS in the event of a master CSS failure, the running-config on the remote CSS must be identical (with some modifications) to the running-config on the local CSS. To automate this configuration synchronization process, you can run the **commit_VipRedundConfig** script on the local CSS to copy the local CSS running-config to the remote CSS running-config.

There are two types of configuration synchronization:

- **Complete** - On CSSs that have an identical chassis (the same CSS model), produces a running-config on the remote CSS that mirrors the running-config on the local CSS.

To perform a complete configuration, run the **script play commit_vip_redundancy** command with the **-a** argument. For more information on running this script and argument, see the "Running the Configuration Synchronization Script" section. Consider running a complete synchronization when you need to replicate interface and circuit configurations including redundant VIPs on a circuit.

This type of synchronization copies reporter configurations for both VRID peering (see the "Configuring VRID Peering" section) and critical phy (see the "Configuring a Critical Physical Interface" section) with the following exceptions:

- If you configure a critical physical interface on a reporter on the local CSS and that interface does not exist on the remote CSS or is of a different interface type (Gigabit Ethernet or Fast Ethernet), the script exits and the synchronization does not complete.
- If you configure one or more VRID IP addresses on a reporter on the local CSS for VRID peering, the script preserves any configured VRID IP addresses on the remote CSS. If the remote CSS configuration does not contain a VRID IP address that corresponds with one in the local CSS configuration, the remote CSS lacks that VRID IP address when the script finishes. The script does copy the reporter configuration from the local CSS to the remote CSS. The script exits with the `File copy Vipr Config Sync Complete` message and indicates that any byte differences between the local and remote configurations exist because the script did not find a corresponding VRID IP address on the remote CSS.

- **Partial** (default) - On CSSs with incompatible configurations, synchronizes all parameter values in the configuration except the interface and circuit configurations including redundant VIPs on a circuit.

  For example, the master is a CSS 11506 and the backup is a CSS 11503. The script maintains the current remote interface and circuit configurations automatically. For CSSs with configured reporters, the script does not copy the reporter configurations to the remote CSS regardless of chassis types.

Script Functions

The configuration synchronization script performs all the necessary steps to update the backup CSS with the master's running configuration. The script:

- Saves the master running-config to the startup-config.
- Archives the startup-config.
- Copies the startup-config to a temporary file (tmp.cfg).
- Calls a function that converts the master VRRP/APP IP addresses to the backup VRRP/APP IP addresses in tmp.cfg.
- If the local VR priority is configured, configures a priority of 100 on the remote VR. If you want to determine mastership based on a different priority, manually configure the remote VR priority as required.
- Uses the **rcmd** command to:
  - Copy tmp.cfg to a temp file on the backup (newconfig)
  - Check newconfig and copy it to the startup-config
  - Clear the backup CSS running-config and script play newconfig

The script performs some verifications before executing the above steps. It checks to see if the local switch is a backup for any VRIDs and asks you if you want to continue, thereby changing the state on the two CSSs. The script also checks the backup to see if it is the master for any VRIDs. If the state is Interface (IF) Down, the script asks you if you want to continue without synchronizing those VRIDs on interfaces that are Down.

Before You Begin

Before you run the configuration synchronization script, ensure that you have configured VIP/interface redundancy and the Application Peering Protocol (APP). For details on configuring VIP/interface redundancy, see the “Configuring VIP and Virtual Interface Redundancy” section. For details on configuring APP, refer to the *Cisco Content Services Switch Global Server Load-Balancing Configuration Guide*.

The synchronization script does not support the following configurations:

- Active/active shared VIP
- Any configuration where some independent VIP addresses are a master while other VIP addresses are a backup

Running the Configuration Synchronization Script

To run the configuration synchronization script, use the **script play commit_vip_redundancy** command in SuperUser mode. The syntax is:

script play commit_vip_redundancy "*arguments***"**

Note When you run the **commit_vip_redundancy** script, if a local VR priority is configured, the script configure a VR priority of 100 on the remote CSS. If you want to determine the mastership based on a different priority, manually configure the remote CSS priority.

By default, the synchronization script synchronizes all parameter values in the configuration except the interface and circuit configurations (including redundant VIPs). When you need to replicate interface and circuit configurations, you can manually configure them on the backup CSS, or consider running the synchronization script with the **-a** argument to replicate them.

You can also run the configuration synchronization script using the predefined alias that comes with all CSSs by entering:

```
# commit_VipRedundConfig "arguments"
```

You can specify the script arguments in any order. The arguments for the commit_vip_redundancy script are:

- *ip address* - The IP addresses of the master and backup APP sessions. This is the only required argument for this script. Use the following syntax when entering the addresses:

  "local *master IP address* **remote** *backup IP address***"**

  For details on automating the entry of the IP address, see "Setting the LOCAL_VIPR_IP and REMOTE_VIPR_IP Variables" later in this section.

- **-a** (All) - Synchronizes the configuration completely. Use this argument only when the master and backup CSSs have identical chassis. This argument synchronizes the entire configuration including the interface and circuit modes. Consider using this argument to replicate interface and circuit configurations including redundant VIPs on a circuit.

  Note Do *not* use this argument if you have different critical services or critical reporters configured on the two CSSs.

- **-d** (Debug) - Debug switch for the commit_vip_redundancy script, which displays the current task being performed as the script progresses. Debug messages display even when you specify the **-s** argument.

Caution Before you use the **-f** argument to remove a config sync lock file, ensure that no one else is running the config sync script on the CSS. Otherwise, if you remove the lock file and then run the script again while the script is in use, the resulting configurations may have some discrepancies.

- **-f** - After an abnormal script termination, removes the lock file so that you can run the script again. This argument overrides all other specified arguments and the script exits immediately after removing the lock file. For details on the lock file, see "Setting the LOCAL_VIPR_IP and REMOTE_VIPR_IP Variables" later in this section.
- **-norlog** (No Remote Log) - Reduces the number of log messages that the CSS sends to the configured log host during the script.
- **-notrap** - Reduces the number of traps that the CSS sends to the configured trap host during the script.
- **-nv** (No Verify) - Informs the script not to verify that the configuration synchronization was successful. However, the script does inform you if the script fails.

  Note By default, the script verifies the configuration synchronization.

- **-s** (Silent) - Suppresses script progress messages and displays only the result of running the script: Commit Successful or Commit Failed. The **-d** argument overrides the **-s** argument.

For example, on the master CSS, run the following script, which uses the defaults of verify on and partial synchronization, plus the IP addresses set as variables and the script alias name:

```
# commit_VipRedundConfig
```

The following output appears:

```
# commit_VipRedundConfig
Verifying app and redundancy configs ...
Checking vip redundancy state ...
Working \
Verifying running-config copy success ...
Commit successful!
```

In this example, the script:

- Performs a partial configuration synchronization (default)
- Verifies that the configuration synchronization was successful (default)

For more information about scripts, refer to the *Cisco Content Services Switch Administration Guide*.

Config Sync Lock File

When you run the script, the software creates a lock file (vipr_config_sync_lock) in the script directory so that you cannot run the script from another session on the CSS. If the lock file exists and you run the script, the following message appears:

```
The script is in use by another session.
```

If the script terminates abnormally, the software does not remove the lock file. The next time you run the script, the above message appears. If you are certain that the script is not in use by another session, use the **-f** argument to remove the lock file.

When you run the script with the **-f** argument, the following message appears and the script exits:

```
VIPR Config Sync lock file removed.
```

Now you can run the script again.

Setting the LOCAL_VIPR_IP and REMOTE_VIPR_IP Variables

To eliminate the need to specify IP addresses each time you run the configuration synchronization script, you can set the value of two variables (LOCAL_VIPR_IP and REMOTE_VIPR_IP) to IP addresses and save them in your user profile. Once you set the variables and save them in your user profile, the variables will always be available after you log in to the CSS. Set the LOCAL_VIPR_IP variable to the IP address of the master CSS and set the REMOTE_VIPR_IP variable to the IP address of the backup CSS.

To set the variables, enter:

```
# set LOCAL_VIPR_IP “master_ip_address” session
# set REMOTE_VIPR_IP “backup_ip_address” session
```

To save the variable in your user profile, enter:

```
# copy profile user-profile
```

Now you can run the configuration synchronization script without typing an IP address.

Note If you already created the MASTER_VIPR_IP and BACKUP_VIPR_IP variables in an earlier release, the script will use the new variables instead, if present.

Logging Configuration Synchronization Script Result Messages

You can specify that script result messages (script success or failure messages) be sent to the current logging device automatically each time you run the configuration synchronization script. To log the script result messages, enable logging on NETMAN with level info-6 or debug-7 by entering:

```
(config)# logging subsystem netman level info-6
```

Note Log messages are generated with or without the -s (silent) argument specified. See the “Running the Configuration Synchronization Script” section.

For example, if the APP session to the backup CSS is not running, the CSS generates the following log message:

```
vipr config sync: app session is DOWN
```

For ease of tracking, each log message contains the string “vipr config sync”.

Displaying VIP and Virtual Interface Redundancy Configurations

The CSS provides **show** commands to enable you to display VIP and virtual interface redundancy configurations. The following sections describe the commands and provide tables describing the output fields.

- Displaying Redundant Virtual Interfaces
- Displaying Redundant VIPs
- Displaying Virtual Router Configurations
- Displaying IP Critical Services
- Displaying Reporter Configurations

Displaying Redundant Virtual Interfaces

To display a list of all redundant virtual interfaces configured on the CSS, use the **show redundant-interfaces** command. This command also displays the status (Enable or Disable) of the DNS server (if configured) on the redundant virtual interface and the number of DNS packets processed by the interface. You may provide an interface IP address option to display only the redundant virtual interfaces present on a particular interface. You may also include a VRID to display only the redundant virtual interface information for a particular VR.

The syntax of this command is:

show redundant-interfaces {*ip_address* {*vrid*}}

The optional variables are:

- *ip_address* - The address for the redundant virtual interface. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1).
- *vrid* - The ID for an existing VR.

For example, to view all redundant interfaces on the CSS, enter:

```
(config) # show redundant-interfaces
```

Table 1-8 describes the fields in the **show redundant-interfaces** command output.

Table 1-8 Field Descriptions for the show redundant-interfaces Command

Field	Description
Interface Address	IP interface address associated with the redundant virtual interface.
VRID	Assigned identifier associated with the VR.
Redundant Address	IP address of the redundant virtual interface.
Range	Not applicable. This field is always set to 1.
State	Current state of the redundant virtual interface. Possible states are: • **Master** - The redundant virtual interface is the master of the flows between the CSS and the network device connected to the redundant virtual interface • **Backup** - The redundant virtual interface is the backup for the flows between the CSS and the network device connected to the redundant virtual interface • **Idle** - The redundant virtual interface is idle
Master IP	IP address of the master VR.
State Changes	Number of times the redundant virtual interface state has changed.
Last Change	Date and time of the redundant virtual interface state last state change.
DNS Server	Status of the DNS server configured on the redundant interface. Possible states are Enable or Disable.
DNS Packets Processed	Number of DNS request packets that the redundant interface has processed.

Displaying Redundant VIPs

To display a list of all redundant VIPs configured on the CSS, use the **show redundant-vips** command. You could provide an interface IP address option to display only the VIPs present on a particular interface. You can also include a VRID to display only the VIP information for a particular VR.

The syntax of this command is:

show redundant-vips {*ip_address* {*vrid*}}

The optional variables are:

- *ip_address* - The address for the redundant interface. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1).
- *vrid* - The ID for an existing VR.

For example, to view all redundant VIPs on the CSS, enter:

```
(config)# show redundant-vips
```

Table 1-9 describes the fields in the **show redundant-vips** command output.

Table 1-9 Field Descriptions for show redundant-vips Command

Field	Description
Interface Address	The IP interface address associated with the redundant VIP.
VRID	The assigned identifier associated with the VR.
Redundant Address	The IP address of the VIP.
Range	The range associated with the VIP.

Table 1-9 Field Descriptions for show redundant-vips Command (continued)

Field	Description
State	Current state of the redundant VIP. Possible states are: • **Master** - The redundant VIP is the master • **Backup** - The redundant VIP is the backup • **Backup Shared** - The redundant VIP is a shared backup • **Idle** - The redundant VIP is idle
Master IP	The IP address of the master VR.
State Changes	The number of times the VIP state has changed.
Last Change	The data and time of the VIP last state change.

Displaying Virtual Router Configurations

To display a list of all VRs configured on the CSS, including their configuration and state information, use the **show virtual-routers** command. If VRID peering (see the "Configuring VRID Peering" section) or critical phy (see the "Configuring a Critical Physical Interface" section) is configured on the CSS, this command also displays any critical reporters associated with the VRs.

You may provide an interface IP address option to display only the VRs present on a particular interface. You may also include a VRID to display only the information for a particular VR.

The syntax of this command is:

show virtual-routers {*ip_address* {*vrid*}}

The optional variables are:

- *ip_address* - The address of the interface. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1).
- *vrid* - The ID for an existing VR.

For example, to view all VRs on the CSS, enter:

```
(config)# show virtual-routers
```

Table 1-10 describes the fields in the **show virtual-routers** command output.

Table 1-10 Field Descriptions for the show virtual-routers Command

Field	Description
Interface Address	The interface IP address associated with the VR.
VRID	The configured identifier of the VR.
Priority	The priority currently being advertised by the VR. Because the priority is dependent on the state of the critical services, the priority may be different than the one configured.
Config. Priority	The configured priority.
State	Current operational state of the VR. Possible states are: • **Master** - The VR is the master. • **Backup** - The VR is the backup. • **Idle** - The VR does not have any redundant virtual interfaces, VIPs, or reporters associated with it. • **Down** - The VR is down. See the Fail Reason field for an explanation of the failure.
Master IP	The IP address of the master VR.
State Changes	The number of times the VR state has changed since the CSS was booted.
Last Change	The data and time of the last VR state change.
Preempt	The state of the preempt option on the VR. If enabled, the state is True; if disabled, the state is False.

Table 1-10 Field Descriptions for the show virtual-routers Command (continued)

Field	Description
Last Fail Reason	The reason that the failover occurred. Codes reported for the Last Fail Reason persist until another failure event occurs. Possible reasons are: • **IF Down** - The IP interface associated with the VR is down. • **Critical Svc Down, Reporter is Down** - One or more critical services and critical reporters associated with the VR are down. • **Critical Phy Down**, **VRID Peering Down** - A critical-phy reporter for the VR and VRID peering are down. • **Critical Svc Down** - One or more critical services associated with the VR are down. • **Critical Phy Down** - A critical-phy reporter for the VR is down. • **VRID Peering Down** - VRID peering is down.
Critical-Services	The names of the critical services associated with the VR.
State	The current condition of the critical service. Possible states are Up, Down, or Suspended.
Type	The type of critical service. Possible types are Scripted, RedundancyUp, or Local.
Critical-Reporters	The names of the critical reporters associated with the VR.

Table 1-10 Field Descriptions for the show virtual-routers Command (continued)

Field	Description
State	The current condition of the critical reporters associated with the VR. Possible states for VRID peering are Master, Backup, Down, or Suspended. Possible states for critical phy are Up, Down, or Suspended.
Type	The type of critical reporter. Possible types are vrid-peering, critical-phy-all-up, or critical-phy-any-up.

Resetting the Virtual Router State Changes Counter

The **show virtual-routers** command displays a State Changes field that records the number of times that a VR changed state since the CSS was booted. To set this counter to zero, use the **zero virtual-router state-changes** command in any mode.

The syntax of this command is:

zero virtual-router state-changes [all|circuit *ip_address* [**all|vrid** *number*]]

The variables and options for this command are:

- **all** - Zeroes the State Changes counter of all VRs configured on the CSS
- **circuit** *ip_address* - Specifies a circuit IP address where VRs are configured
- **all** - Zeroes the State Changes counter of all VRs on the specified circuit
- **vrid** *number* - Zeroes the State Changes counter of the specified VR on the specified circuit

For example, to reset the State Changes counter for all VRs configured on the circuit with an IP address of 192.168.1.7, enter:

```
(config)# zero virtual-router state-changes circuit 192.168.1.7 all
```

Displaying IP Critical Services

To display a list of all critical services configured on the CSS, use the **show critical-services** command. You can provide an interface IP address option to display only the critical services present on a specific interface. You may also include a VRID to display only the critical service information for a specific VR.

The syntax of this command is:

show critical-services {*ip_address* {*vrid*}}

The optional variables are:

- *ip_address* - The address for the redundant interface. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1).
- *vrid* - The ID for an existing VR.

For example, to view all critical services on the CSS, enter:

```
# show critical-services
```

Table 1-11 describes the fields for the **show critical-services** command output.

Table 1-11 Field Descriptions for the show critical-services Command

Field	Description
Interface Address	The IP interface address associated with the VR.
VRID	The assigned identifier associated with the VR.
Service Name	The name of the critical service.

Table 1-11 Field Descriptions for the show critical-services Command (continued)

Field	Description
Service Type	The type of critical service. Possible critical service types are: • **Scripted** - A service whose state depends upon a running script or a named keepalive. • **Redundancy-up** - A service whose state depends upon the state of an ICMP keepalive on a router. • **Local** - Every type of service other than a scripted service or a redundancy uplink service. Typically, this is a Web server.
Service State	The current state of the critical service. The State field displays the service as either Alive, Dying, Down, or Suspended. The Dying state reports that a service is failing according to the parameters configured in the following service mode commands: **keepalive retryperiod**, **keepalive frequency**, and **keepalive maxfailure**. When a service enters the Down state, the CSS does not forward any new connections to it (the service is removed from the load balancing rotation for the content rule). However, the CSS keeps all existing connections to the service (connections to that service are not torn down).

Displaying Reporter Configurations

To display reporter configurations for VRID peering and critical phy, use the following commands:

- **show reporter** {*reporter_name*|**summary**} - Displays the reporter name, type, state, and the critical interfaces and their link states.
- **show running-config reporter** - Displays the configurations of all reporters configured on the CSS.
- **show running-config reporter** *reporter_name* - Displays the configuration of the specified reporter.
- **show virtual-routers** - Displays the configured virtual router information and the critical reporter state and type. See the "Displaying Virtual Router Configurations" section.
- **show critical-reporters** - Displays the critical reporters configured for critical phy.
- **show running-config circuit** - Displays the critical reporter associated with a circuit VLAN for a critical phy.

Using the show reporter Command

To display reporter configuration and statistics for VRID peering or critical phy, use the **show reporter** command. The syntax of this command is:

show reporter {*reporter_name*|**summary**}

The variables and options for this command are:

- *reporter_name* - Name of an existing reporter
- **summary** - Displays summary statistics for all configured reporters

Table 1-12 describes the fields for the **show reporter** command output.

Table 1-12 Field Descriptions for the show reporter Command

Field	Description
Name	Name of the reporter whose configuration you are displaying.
State	Current state of the reporter. Possible reporter states for VRID peering are Master, Backup, Suspended, or Down. Possible reporter states for critical phy are Up, Suspended, or Down.
Type	Type of reporter. Possible reporter types are: • vrid-peering (see the "Configuring VRID Peering" section) • critical-phy-all-up (see the "Configuring a Critical Physical Interface" section) • critical-phy-any-up (see the "Configuring a Critical Physical Interface" section)
State Transitions	Number of times the reporter state changed since the last time the CSS was booted. If the State Transitions field is 0, the 0 value can be due to a counter reset through the global configuration mode **zero reporter state-transitions** command. The counter can also be 0 if the reporter is down.
Circuit (VRID peering only)	Circuit IP address of the CSS.
VRID (VRID peering only)	Identifier of the VR associated with the reporter.
State (VRID peering only)	Current state of the VR associated with the reporter. Possible states of the VR are Master, Backup, Down, or Unknown.
Interface (critical phy only)	Interfaces associated with the reporter.
Link (critical phy only)	Current state of the physical link interface. Possible link states are Up or Down.

Resetting the Reporter State Transitions Counter

The **show reporter** command displays a State Transitions field that records the number of times that a reporter changed state since the CSS was booted. To set this counter to zero, use the **zero reporter state-transitions** command in any command mode.

The syntax of this command is:

zero reporter state-transitions [**all**|**reporter** *reporter_name*]

The variables and options for this command are:

- **all** - Zeroes the State Transitions counter of all reporters configured on the CSS
- **reporter** *reporter_name* - Zeroes the State Transitions counter of the specified reporter

For example, to reset the State Transitions counter for reporter r1, enter:

```
(config)# zero reporter state-transitions reporter r1
```

Displaying a Reporter Configuration in the Running-Config

To display a reporter configuration in the running-config, use the **show running-config reporter** command in any mode. This command displays all reporter configurations on the CSS. To display a specific reporter configuration, use the **show running-config reporter** *reporter_name* command.

Displaying Critical Reporter Information

To display critical reporter configuration information for VRID peering and critical phy, use the **show critical-reporter** command in any mode. The syntax of this command is:

show critical-reporters *ip_address vrid*

The variables for this command are:

- *ip_address* - Specifies the interface address of the virtual router associated with the critical-reporter
- *vrid* - Specifies the VRID of the VR associated with the critical reporter

Table 1-13 describes the fields for the **show critical-reporters** command output.

Table 1-13 Field Descriptions for the show critical-reporters Command

Field	Description
Interface Address	IP address of the VR interface.
VRID	Identifier of the VR associated with the reporter.
Reporter Name	Name of the reporter with which the VR is associated.
Reporter Type	Configured type of the reporter. Possible types are vrid-peering, critical-phy-all-up, or critical-phy-any-up.
State	State of the reporter. Possible states are Master, Backup, Up, Down, or Suspended.

Displaying Critical Reporters in the Running-Config

To display configured critical reporters in the running-config, use the **show running-config circuit** command. This command displays all configured circuits and any critical reporters associated with a virtual router.

Configuring Adaptive Session Redundancy

This chapter describes how to configure Adaptive Session Redundancy (ASR) for stateful failover on a CSS.

This chapter contains the following major sections:

- Overview of CSS Redundancy
- Configuring Adaptive Session Redundancy
- Displaying ASR Information

Overview of CSS Redundancy

Redundancy helps to ensure:

- High availability for your network applications
- Users do not experience long network delays or black holes due to a single point of failure.

A CSS provides three types of redundancy.

- Virtual IP (VIP) redundancy and virtual interface redundancy - Provide redundant VIP addresses and redundant virtual interfaces for fate sharing and server default gateways. For details, see Chapter 1, Configuring VIP and Virtual Interface Redundancy.
- Adaptive Session Redundancy (ASR) - Provides session-level redundancy (stateful failover) to continue active flows without interruption if the master CSS fails over to the backup CSS. For details, see this chapter.
- Box-to-box redundancy - Provides chassis-level redundancy between two identically configured CSSs. For details, see Chapter 3, Configuring Box-to-Box Redundancy.

The following sections provide information about when (and when not) to use the different types of redundancy.

When to Use VIP and Virtual Interface Redundancy

Typically, you configure VIP redundancy on the public side of CSS peers that are positioned in front of a server farm. You configure virtual interface redundancy on the private-side interfaces attached to the L2 device in front of the servers.

Configure VIP redundancy:

- With virtual interface redundancy to provide fate sharing
- When you have a common subnet between the two CSSs on which the VIPs reside
- As a prerequisite to configuring ASR (requires active-backup VIP redundancy)
- To provide active-active CSS behavior (both CSSs processing flows)

Configure interface redundancy:

- With VIP redundancy to provide fate sharing
- When you need a default gateway for the back-end servers
- Instead of VIP redundancy on the client side of the CSS when the VIPs are on a subnet different from the subnet of your uplinks

When to Use ASR

ASR provides session-level redundancy for applications where active flows (including TCP and UDP) must continue without interruption, even if the master CSS fails over to the backup CSS.

Configure ASR:

- If you require stateful failover for mission-critical applications (for example, enterprise applications, long-lived flows, such as HTTP or FTP file transfers, and e-commerce)
- After you have first configured active-backup VIP and virtual interface redundancy

When to Use Box-to-Box Redundancy

Configure box-to-box redundancy when you:

- Expect the behavior of the CSSs to be active/standby (only the master CSS processes flows)
- Can configure a dedicated Fast Ethernet (FE) link between the CSSs for the VRRP heartbeat

Do not configure box-to-box redundancy when you:

- Expect the behavior of the CSSs to be active-active (both CSSs processing flows). Use VIP redundancy instead.
- Cannot configure a dedicated FE link between the CSSs.

Configuring Adaptive Session Redundancy

Configure Adaptive Session Redundancy (ASR) on Cisco 11500 series CSS peers in an active-backup VIP redundancy and virtual interface redundancy environment to provide stateful failover of existing flows. ASR ensures that, if the master CSS fails, the backup CSS has the necessary flow-state information to continue any active flows (including TCP and UDP) without interruption when the backup CSS assumes mastership. “Adaptive” means that you can configure ASR on a per content rule basis.

Use ASR for:

- Mission-critical enterprise applications.
- Long-lived flows such as FTP and HTTP file transfers.
- E-commerce applications, such as online stock trading or banking where users must remain connected to a service for the duration of a transaction even if the master CSS fails.

In an ASR configuration, CSSs replicate flows that are:

- Fully-resolved (the master CSS has received a SYN/ACK from a server)
- Set up using content rules, services, and source groups that you specify as *redundant*

Note For implicit or explicit Layer 5 rules, where there is delayed binding, binding is not complete until the CSS processes the SYN/ACK from the server. If a failover occurs in the middle of a spanned content request, the master CSS will not receive the SYN/ACK from the server and the flow will not be replicated on the backup CSS. No data is lost and users can simply refresh their browsers to restart the connection.

Note During an FTP failover, the control channel and/or the data channel need to share information with the backup CSS. If the current state information has not been fully transferred across the ISC link to the backup CSS, then the flow may be lost.

This section includes the following topics:

- Stateful Failover
- Inter-Switch Communications
- Redundant Indexes
- Configuration Requirements and Restrictions
- ASR Quick Start
- Configuring Inter-Switch Communications
- Configuring Redundant Services
- Configuring Redundant Content Rules
- Configuring Redundant Source Groups
- Source Group Port-Mapping Behavior in an ASR Configuration
- Synchronizing ASR Configurations

Stateful Failover

Active flows that match a redundant content rule, service, or source group on the master CSS are replicated as *dormant flows* on the backup CSS peer. A dormant flow contains all the flow-state information necessary for the backup CSS to take over the flow if the master CSS fails, including the flow ID assigned by the session processor (SP) that created the flow. If the master CSS fails, the dormant flows on the backup CSS become active when the backup CSS assumes mastership of the VIP. In turn, the active flows on the former master CSS transition to a dormant state to fully back up the active flows on the new master CSS.

A master CSS maps a newly activated TCP flow after it receives the first packet for the flow. If it can resolve a single route back to the source address, a CSS attempts to map a UDP flow when it activates the flow. Otherwise, the CSS maps the UDP flow after it receives the first packet of the flow.

Inter-Switch Communications

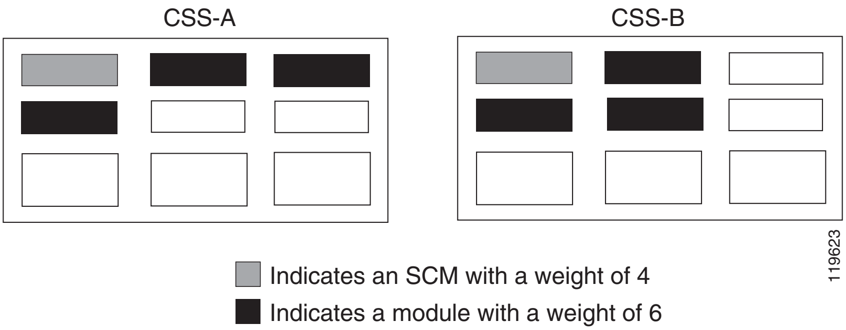
In an ASR configuration, CSS peers share redundant flow-state information over a maximum of two private Inter-Switch Communications (ISC) links after booting. ISC is a messaging service used by CSSs to exchange flow-state information. Only one ISC link is active at a time. The other ISC link (if configured) remains in backup mode until needed.

To prevent incorrect flow and port mapping information from being replicated to the backup CSS, a CSS does not activate its ISC link if it detects a mismatched chassis configuration with the other CSS. During its discovery phase and before activating the link, the ISC protocol exchanges chassis information between the two CSSs to ensure that:

- The two chassis have the same number of modules session processors (SPs) with the same weights. SCMs have a weight of 4; all other CSS modules have a weight of 6. To display the weights of the modules in your CSS, enter the **show chassis session-processors** command.
- The modules (SPs) are installed in the same order. Skipping slots is permitted provided that the overall order is the same and the CSS can match up all the SPs in both chassis.

For example, Figure 2-1 shows two CSS 11506s with slightly different module installation configurations. Each CSS 11506 has four installed modules, but the CSS-B configuration skips slot 3. As far as ISC is concerned, the configurations match because both CSSs have the same number of SPs with the same weights and installed in the same overall order.

Figure 2-1 Example of CSS 11506 Matching Module Configurations for ISC


CSS-A
CSS-B
119623
Indicates an SCM with a weight of 4
Indicates a module with a weight of 6

To determine if an ISC link is up, a CSS uses a mechanism called LifeTick. LifeTick sends an asynchronous message that contains information about the selected path. If the CSS does not receive a LifeTick message within two seconds, the CSS considers the ISC link to be down. If a second link is configured, the CSS uses that link for ISC.

Note For best results, we recommend that you use the Gigabit Ethernet (GE) ports for the ISC links in all cases. If you are using a CSS 11501, use the GE port for ISC and the Fast Ethernet (FE) ports for normal traffic.

The ISC links use the GE ports or the FE ports on the CSS session processors (SPs) to send ISC messages containing flow-state information. Once you configure the ISC ports, those ports are dedicated to ISC and you cannot use those same ports for non-ISC traffic.

Note You must connect the ISC ports directly to the two CSSs. You cannot use Layer 2 devices on the ISC links between the two CSSs. Also, the ISC links must be dedicated to passing only ISC traffic.

For new flows, CSSs exchange flow states in real time over the ISC links. For existing flows, CSSs exchange flow states at bootup and at VIP redundancy failover.

Redundant Indexes

ASR uses unique global redundant indexes to keep track of content rules, services, and source groups configured on the redundant CSS peers. Set up the redundant indexes in rules, services, and groups using the **redundant-index** command. You must then configure identical redundant content rules, services, and source groups on CSS peers in the ASR configuration.

Each redundant index that you configure on a rule, service, or group must be unique among all rules, services, or groups configured on a redundant pair of CSSs. For example, if you configure a rule with a redundant index of 1 on a pair of CSSs, you cannot configure an index of 1 on another rule. However, you could configure an index of 1 on a group or service if that value has not already been used on a group or a service.

Note If you run traffic to a configuration that contains discrepancies between the redundant indexes on the two CSSs, the CPU utilization for each processor on the CSS may climb to an abnormal level (at 2000 flows/second, approximately 50 percent utilization for each processor). If you set the logging level to notice-5 or higher, the SCM utilization may peak at approximately 90 percent because each connection generates a redundant index mismatch log entry. For example: AUG 7 14:12:15 3/1 1124272 SLR-5: Rejected. Redundant global rule index (7) not found.

Configuration Requirements and Restrictions

The following requirements and restrictions apply to both CSS peers in an ASR configuration:

- Ensure that both CSSs have the same number of SPs. Otherwise, the CSSs cannot activate the ISC link.
- You cannot configure ASR and an SSL module on the same service. ASR does not support the replication of flows to an SSL module in the CSS.
- You cannot configure ASR and HTTP compression on the same service. The CSS does not allow you to configure a redundant index on a content rule that has services with compression enabled or to enable compression on a service in a rule that is configured with a redundant index.
- Configure VIP and virtual interface redundancy on both CSS peers. For details, refer to Chapter 1, Configuring VIP and Virtual Interface Redundancy.
- Configure a redundant VIP in a redundant content rule or source group. To activate a redundant content rule or source group, you must associate the rule or group with a redundant VIP.
- Ensure that VIP ranges specified in redundant content rules and source groups are the same as the VIPs associated with virtual routers for VIP redundancy. If the redundant content rule or source group VIPs are a superset, ASR is supported only for the VIPs that are associated with the virtual routers. For the remaining VIPs, the behavior is undefined when a failover occurs, because it is unclear whether those VIPs are mastered on the new master CSS or not.

- You cannot configure VIP wildcard or double-wildcard caching rules because they do not require a VIP. For information on wildcard cache rules, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.
- Configure ISC on both CSSs. This allows the CSSs to share flow-state information.
- Configure a maximum of two ISC ports on a CSS. Multiple ports must reside on the same module in the CSS 11503 or CSS 11506 or on the same CSS 11501. Also, the ports must be of the same type (Gigabit Ethernet or Fast Ethernet) in both CSSs. For best results, we recommend that you use GE ports for the ISC links in all cases.
- Ensure that the ISC ports are not configured in any VLANs. If necessary, remove the designated ports from all VLANs before configuring ISC. For details on disabling an interface port from a VLAN, refer to the *Content Services Switch Routing and Bridging Configuration Guide*.
- You must connect the ISC ports directly to the two CSSs. You cannot use Layer 2 devices on the ISC links between the two CSSs. Also, the ISC links must be dedicated to passing only ISC traffic.
- If you configure any ISC ports on an SCM, you can have only one SCM installed in the CSS 11506.
- The CSS 11501 does not support redundant GE ISC links for ASR because the CSS includes only a single GBIC port.
- Ensure that any service configured with connection limits, marked as redundant, *and* used by at least one redundant content rule is used only by other content rules that are also redundant. If this is not true, there could be redundant and nonredundant flows connected to the service with connection limits.

  In case of a failover, no information is available for the nonredundant flows on the backup CSS. Until the server cleans up the nonredundant flow connections, they continue to contribute to the connection limit on the service without the backup CSS having any knowledge of how many such connections exist. Making all flows redundant by imposing the above restrictions eliminates this problem.
- When you configure critical services, be sure to change the default keepalive settings to the following recommended settings for ASR. For example, enter:

  ```
  service CriticalService
  ```

```
ip address 192.168.2.1
keepalive frequency 2
keepalive maxfailure 2
keepalive retryperiod 2
active
```

Note The above keepalive values are a recommended starting point. Some scripted keepalives may take longer than two seconds to run. You may need to adjust your keepalive values so that the CSS detects a failure before your application times out.

- Configure as redundant any source groups that you specify in ACL clauses. Otherwise, ASR does not require source groups. It is helpful to configure ACLs similarly on the master and backup CSSs. This ensures that the CSSs share the port-map state during flow setup time, and, at failover time, a CSS finds the same ACL and source group configured on the peer. Otherwise, when a flow fails over to the backup, it is possible that the flow may match on a different ACL clause that has no source group configured or a different source group (possibly a nonredundant one).

  Source groups selected by ACL-checking always take precedence over other source group matches for a flow. Therefore, if the master and backup CSSs have different ACL definitions, when a flow fails over to the backup and the source group selected on the master is not found on the backup, the CSS rejects the flow. Also, if the flow matches on a different source group through an ACL, that source group takes precedence over the redundant source group that was sent from the master.
- Configure as redundant any preferred service that you configured in an ACL clause.
- Configure mutually exclusive port-map ranges on the redundant peers using the **global-portmap** command to avoid potential network port collisions. Keeping the port-map ranges mutually exclusive on the redundant peer also eliminates the need to dynamically update the global port-map database on the backup CSS. For more information on port mapping, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.
- Adaptive Session Redundancy (ASR) imposes restrictions on the number of available and eligible source ports in a source group because of the mapping of resources to the backup CSS with an unknown chassis configuration. For details, see the "Source Group Port-Mapping Behavior in an ASR

Configuration" section. For more information about source groups, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

- Do not configure ASR and stateless redundancy failover on the same CSS. Such a configuration is not supported. For details on stateless redundancy failover, refer to Chapter 3, Configuring Box-to-Box Redundancy, in the "Configuring Stateless Redundancy Failover" section.
- ASR does not support NAT peering. For details on NAT Peering, refer to the *Content Services Switch Content Load-Balancing Configuration Guide*.

Upgrading to WebNS Version 7.40 and Higher

To maximize the number of ports available for PAT in an ASR configuration, the CSSs must have similar chassis configurations in terms of the total number of session processors (SPs) and their assigned weights. All CSS module types have the same assigned weights, except for the SCM. The SSL module and the backup SCM in a dual SCM configuration are not considered SPs. Therefore, in an ASR configuration, both CSSs must have the same number of SPs.

If you are upgrading from a version of WebNS software earlier than 7.40, be aware of the following ASR configuration restrictions in WebNS software versions 7.40 and higher:

- If your CSSs have mismatched chassis configurations, ASR does not work after the upgrade
- If your CSSs meet the ASR requirement of having the same number of SPs in each chassis, you must upgrade both CSSs to WebNS Version 7.40
- During the upgrade process, ASR does not work and you lose any sessions that are in progress

ASR Quick Start

Table 2-1 provides a quick overview of the steps required to configure ASR for *each* CSS in the redundant configuration. Each step includes the CLI command or a reference to the procedure required to complete the task. For a complete description of each feature and all the options associated with the CLI command, see the sections following Table 2-1.

Table 2-1 ASR Configuration Quick Start

Task and Command Example

1. Enter config mode.
`# config`
`(config)#`

2. Configure active/backup VIP and virtual interface redundancy. Refer to Chapter 1, Configuring VIP and Virtual Interface Redundancy earlier in this chapter.

3. Configure a maximum of two directly connected (no intervening L2 devices) ISC links on Gigabit Ethernet or Fast Ethernet ports between the two redundant CSSs. See “Configuring Inter-Switch Communications” later in this chapter.
`(config)# interface 1/1`
`(config-if[ 1/1])# isc-port-one`
`(config)# interface 1/2`
`(config-if[ 1/2])# isc-port-two`

4. Configure services that are targets of redundant content rules. For more information on services, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.
`(config)# service server1`
`(config-service[server1])# ip address 192.168.100.100`
`(config-service[server1])# redundant-index 1`
`(config-service[server1])# active`

5. Configure redundant content rules and add the redundant services. For more information on content rules, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.
`(config)# owner arrowpoint`
`(config-owner[arrowpoint])# content rule1`
`(config-owner-content[arrowpoint-rule1]# vip address 192.1.1.100`
`(config-owner-content[arrowpoint-rule1]# protocol tcp`
`(config-owner-content[arrowpoint-rule1]# port 80`
`(config-owner-content[arrowpoint-rule1]# url “/redundant.html”`
`(config-owner-content[arrowpoint-rule1]# add service server1`
`(config-owner-content[arrowpoint-rule1]# redundant-index 5`
`(config-owner-content[arrowpoint-rule1]# active`

Table 2-1 ASR Configuration Quick Start (continued)

Task and Command Example

6. Configure redundant source groups and add the redundant services. For more information on source groups, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

```
(config)# group group1
(config-group[group1])# vip address 192.1.1.100
(config-group[group1])# add service server1
(config-group[group1])# redundant-index 4
(config-group[group1])# active
```

7. Configure global port mapping (port translation) with mutually exclusive port ranges on the CSS peers to avoid potential port collisions. For more information on CSS port mapping, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

   For example, on one CSS peer, enter:

```
(config)# global-portmap base-port 3000 range 30000
```

   On the other CSS peer, enter:

```
(config)# global-portmap base-port 33100 range 30000
```

8. Configure the same redundant services, content rules, and source groups on the other CSS peer (synchronize the configurations).

9. Use the **show session-redundant all** command to verify your configuration.

```
# show session-redundant all
```

The following running-config example shows the results of entering the commands in Table 2-1 (in bold text) and the commands used to configure VIP and virtual interface redundancy (see Chapter 1, Configuring VIP and Virtual Interface Redundancy).

```
!************************** GLOBAL ***************************
  global-portmap base-port 3000 range 30000

!************************ INTERFACE ************************
interface  1/1
  isc-port-one
```

```
interface  1/2
  isc-port-two

interface  2/1
  bridge vlan 2

!************************* CIRCUIT *************************
circuit VLAN1

  ip address 10.1.1.1 255.255.255.0
    ip virtual-router 1 priority 101 preempt
    ip redundant-interface 1 10.1.1.254
    ip critical-service 1 upstream_downstream

circuit VLAN2

  ip address 192.1.1.1 255.255.255.0
    ip virtual-router 2 priority 101 preempt
    ip redundant-vip 2 192.1.1.100
    ip critical-service 2 upstream_downstream

!************************* SERVICE *************************
service server1
  ip address 10.1.1.50
  redundant-index 1
  active

service upstream_downstream
  ip address 192.1.1.50
  keepalive type script ap-kal-pinglist “192.1.1.20 10.1.1.20"
  keepalive frequency 2
  keepalive maxfailure 2
  keepalive retryperiod 2
  active

!*************************** OWNER ***************************
owner arrowpoint

  content rule1
    vip address 192.1.1.100
    protocol tcp
    port 80
    url “/redundant.html”
    add service server1
    redundant-index 5
    active
```

```
!************************** GROUP **************************
group group1
  vip address 192.1.1.100
  add service server1
  redundant-index 4
  active
```

Configuring Inter-Switch Communications

Inter-Switch Communications (ISC) is a messaging service that CSS peers use to exchange flow-state information in an ASR configuration. If the master CSS fails, the backup CSS already has the flow-state information necessary to continue the current flows without interruption. Using ISC, CSSs exchange state information:

- For existing flows at boot-up time and at VIP redundancy failover
- For new flows in real time (after the CSS receives a SYN/ACK from the server)

To prevent incorrect flow and port mapping information from being replicated to the backup CSS, the CSS does not bring up the ISC link if it detects a mismatched chassis configuration with the other CSS. During the discovery phase, the ISC protocol exchanges chassis information between the two CSSs and ensures that both chassis have the same number of SPs before bringing up the link.

To enable ISC between two CSSs in an ASR configuration, use the **isc-port-one** and **isc-port-two** commands in interface configuration mode. You can configure a maximum of two ISC ports on each CSS. The two ports must be of the same type (Gigabit Ethernet or Fast Ethernet) and must be on the same module in the CSS 11503 or CSS 11506 or on the same CSS 11501. When you configure two ISC ports, the first port is active and the second port remains in a backup state. The backup link is used only if the active link fails. For best results, we recommend that you configure ISC on the GE ports.

The CSS 11501 does not support redundant GE ISC links for ASR because that CSS model includes only one GE port.

You must connect the ISC ports directly to the two CSSs. You cannot use Layer 2 devices on the ISC links between the two CSSs. Also, the ISC links must be dedicated to passing only ISC traffic.

For example, to enable both ISC ports on a CSS 11506, enter:

```
(config)# interface 1/1
(config-if[ 1/1])# isc-port-one
(config-if[ 1/1])# interface 1/2
(config-if[ 1/2])# isc-port-two
```

To disable both ISC ports on a CSS 11506, enter:

```
(config)# interface 1/1
(config-if[ 1/1])# no isc-port-one
(config-if[ 1/1])# interface 1/2
(config-if[ 1/2])# no isc-port-two
```

Configuring Redundant Services

To configure the global service index for a redundant service, use the **redundant-index** command. A CSS uses the global service index to keep track of redundant services and associated flow-state information.

The syntax for this service configuration mode command is:

redundant-index *index*

The variable *index* is a unique number you assign to a redundant service. Enter a unique integer from 0 to 32767, where a value of 0 disables ASR for a service. The default is 0, but it does not appear in the running-config even if you configure it explicitly.

For example:

```
(config-service[server1])# redundant-index 5
```

To disable ASR for a service, enter:

```
(config-service[server1])# no redundant-index
```

Note If you issue the **no redundant-index** command on an active redundant service for live redundancy peers, the command automatically suspends the service. Flows already mapped by a CSS are not affected. However, if a failover occurs during the life of an active flow that matches on such a suspended service, the backup CSS cannot map the flow because it cannot find the service with the same global index as that on the original master.

For more information on configuring services, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

Configuring Redundant Content Rules

To configure the global content index for a redundant content rule, use the **redundant-index** command. A CSS uses the global content index to keep track of redundant content rules and associated flow-state information.

The syntax for this content configuration mode command is:

redundant-index *index*

The variable *index* is a unique number you assign to a redundant content rule. Enter a unique integer from 0 to 32767, where a value of 0 disables ASR on a content rule. The default is 0, but it does not appear in the running-config even if you configure it explicitly.

For example:

```
(config-owner-content[arrowpoint-rule1]# redundant-index 1
```

To disable ASR on a content rule, enter:

```
(config-owner-content[arrowpoint-rule1]# no redundant-index
```

Note If you issue the **no redundant-index** command on an active redundant content rule for live redundancy peers, the command automatically suspends the content rule. Flows already mapped by a CSS are not affected. However, if a failover occurs during the life of an active flow that matches on such a suspended content rule, the backup CSS cannot map the flow because it cannot find the content rule with the same global index as that on the original master.

For more information on configuring content rules, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

Configuring Redundant Source Groups

If you configured source groups in ACLs, you must configure those source groups as redundant. Otherwise, ASR does not require source groups. To configure the global source group index for a redundant source group, use the **redundant-index** command. A CSS uses the global source group index to keep track of redundant content rules and associated flow-state information.

The syntax for this group configuration mode command is:

redundant-index *index*

The variable *index* is a number you assign to a redundant source group. Enter a unique integer from 0 to 32767, where a value of 0 disables ASR for a source group. The default is 0, but it does not appear in the running-config even if you configure it explicitly.

For example, to enable ASR for a source group:

```
(config-group[group1])# redundant-index 4
```

To disable ASR for a source group, enter:

```
(config-group[group1])# no redundant-index
```

Note If you issue the **no redundant-index** command on an active redundant source group on live redundancy peers, the command automatically suspends the source group. Flows already mapped by a CSS are not affected. However, if a failover occurs during the life of an active flow that matches on such a suspended source group, the backup CSS cannot map the flow because it cannot find the source group with the same global index as that on the original master.

For more information on configuring source groups, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

Source Group Port-Mapping Behavior in an ASR Configuration

Because Adaptive Session Redundancy (ASR) requires that both CSSs have the same number of SPs in each chassis, each CSS uses the same port-selection algorithm in an ASR configuration as in a non-ASR configuration. This behavior means that ASR imposes no further restrictions on source group port mapping. For more information about source groups and port mapping, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

Synchronizing ASR Configurations

You must synchronize configurations on both CSS peers to ensure that the ASR-specific configurations on the master CSS and the backup CSS are the same. This is critical to the proper functioning of ASR.

For ASR, you must manually configure on each peer:

- ISC
- Redundant content rules
- Redundant services
- Redundant source groups

Displaying ASR Information

Use the commands described in the following sections to display information for:

- Inter-Switch Communications (ISC)
- Dormant flows
- ASR status and global redundant indexes

Displaying Inter-Switch Communications Ports

Use the **show isc-ports** command to display the following information:

- Ports configured for ISC on a CSS
- Status of the ISC link
- Reason the ISC link failed if it is down

Table 2-2 describes the fields in the **show isc-ports** output.

Table 2-2 Field Descriptions for the show isc-ports Command

Field	Description

Inter-Switch Communications Configuration	Lists the CSS ports (in slot/port format) configured for ISC port one and ISC port two. If ISC is not configured, the command displays the following messages:
	`Inter-Switch Port One is not configured.`
	`Inter-Switch Port Two is not configured.`

Inter-Switch Communications Status	Indicates whether ISC is Up or Down and, if Up, on which CSS port ISC is currently active.
Port # Communication Failure Reason	Indicates why the ISC link failed. Use this field to help you troubleshoot the ISC link if it fails. Possible reasons are: • None - ISC link is up. • No Interface Assigned - An interface was not assigned to the ISC port. • No Physical Link - There is no physical link on the ISC port. • No Discovery Response - The remote CSS did not respond to the Hello message from the local CSS during the discovery phase of the ISC protocol. • Wrong Protocol Version - Different ISC protocol versions are running on the two CSSs. • Mismatched Chassis - The CSSs have different chassis configurations (different number of SPs).

Displaying Dormant Flow Information

To display information about the current dormant flows on the backup CSS in an ASR configuration, use the **show dormant flows** command. Dormant flows are flows on the backup CSS that become active if the master CSS fails and the backup CSS becomes the master.

The syntax for this command is:

show dormant flows {*source_address* {*destination_address*}}

The optional variables for this command are:

- *source_address* - Displays dormant flows for the specified source IP address. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1).
- *destination_address* - Displays dormant flows for the specified destination IP address. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1).

Table 2-3 describes the fields in the **show dormant flows** output.

Table 2-3 Field Descriptions for the show dormant flows Command

Field	Description
Src Address	The source address for the flow.
SPort	The source port for the flow.
Dst Address	The destination address for the flow.
DPort	The destination port for the flow.
NAT Dst Address	The network address translation (NAT) destination address.
Prt In	Not applicable. A dormant flow does not have a port associated with it.
OutPort	Not applicable. A dormant flow does not have a port associated with it.

To display summary information about redundant dormant flows, use the **flow statistics dormant** command.

Table 2-4 describes the field in the **flow statistics dormant** output.

Table 2-4 Field Descriptions for the flow statistics dormant Command

Field	Description
Redundant Flow Statistics - Slot *n*, Subslot *n*	Inactive redundant flow statistics for the module in the specified slot and subslot in the backup CSS.
Dormant Flow Count	Total number of inactive redundant flows in the specified module in the backup CSS.
UDP Flows	Number of inactive redundant UDP flows in the specified module in the backup CSS.
TCP Flows	Number of inactive redundant TCP flows in the specified module in the backup CSS.
Redundant Flow Statistics - Aggregate	Total Inactive redundant flow statistics for all of the modules in the backup CSS.
Total UDP Flows	Total number of inactive redundant UDP flows in the backup CSS.
Total TCP Flows	Total number of inactive redundant TCP flows in the backup CSS.
Total Flows	The total number of inactive redundant flows in the backup CSS from active redundant flows on the master CSS. The dormant flows contain all the flow-state information necessary for the backup CSS to master the flows if the master CSS fails. If the master CSS fails, the backup CSS becomes the master CSS and the dormant flows become active flows.

Displaying ASR Information for Content Rules, Services, and Source Groups

The following sections describe how to display ASR information specific to content rules, services, and source groups.

Displaying ASR Status and Global Index Values

To display information about ASR status and global redundant indexes, use the following commands:

- **show rule**
- **show service**
- **show group**

The relevant fields in the output of these commands are:

- **Session Redundancy** - The state of ASR for the content rule, service, or source group. Possible values are: Enabled or Disabled
- **Redundancy Global Index** - The unique global index value for ASR configured for the content rule, service, or source group using the **redundant-index** command.

For complete details on the **show rule**, **show service**, and **show group** commands, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

Displaying Summary ASR Information

To display summary ASR information about redundant content rules, services, and source groups, use the **show session-redundant** command.

The syntax for this global configuration mode command is:

show session-redundant [**rule**|**service**|**group**|**all**]

The optional keywords are:

- **rule** - Displays summary ASR information for redundant content rules.
- **service** - Displays summary ASR information for redundant services.

- **group** - Displays summary ASR information for redundant source groups.
- **all** - Displays summary ASR information for content rules, services, and source groups.

For example, to view summary ASR information for redundant content rules, enter:

```
(config)# show session-redundant rule
```

Table 2-5 describes the fields.

Table 2-5 Field Descriptions for the show session-redundant Command

Field	Description
Session Redundant Content Rules	
Content Rule	The redundant content rule name.
Content Rule State	The current state of the redundant content rule. Possible states are: Active or Suspend.
VIP Address	The virtual IP address of the redundant content rule in dotted decimal notation.
Redundancy Global Index	The ASR global index configured for the redundant content rule.
Redundancy State	The state of the CSS peer: Master, Backup, or Suspend.
Rule Redundant Services 1	The name of the redundant service and its global index value configured on the rule.
Session Redundant Services	
Service	The name of the redundant service.
Service State	The current state of the redundant service. Possible states are: Alive, Dying, or Down.
IP Address	The virtual IP address of the redundant service in dotted-decimal notation.
Redundancy Global Index	The ASR global index configured for the redundant service.

Table 2-5 Field Descriptions for the show session-redundant Command (continued)

Field	Description
Session Redundant Source Groups	
Source Group	The name of the redundant source group.
Source Group State	The current state of the redundant source group. Possible states are: Active or Suspend.
VIP Address	The virtual IP address of the redundant source group.
Redundancy Global Index	The ASR global index configured for the redundant source group.
Group Redundant Services	
Source Services	The redundant source services configured in this redundant source group, their keepalive state, and global index. If no source services are configured in this source group, the value is NONE.
Destination Services	The redundant destination services configured in this redundant source group and their keepalive state. If no destination services are configured in this source group, the value is NONE.

CHAPTER 3

Configuring Box-to-Box Redundancy

This chapter describes how to configure redundancy between two identically configured Cisco Content Services Switches (CSSs). Information in this chapter applies to all CSS models, except where noted.

This chapter contains the following major sections:

- Overview of CSS Redundancy
- Redundancy Protocol Overview
- Redundancy Configuration Quick Start
- Cabling Redundant CSSs
- Configuring Redundancy
- Synchronizing a Redundant Configuration
- Using the Redundancy Force-Master Command
- Configuring Multiple Redundant Uplink Services
- Using the redundancy-phy Command
- Configuring Stateless Redundancy Failover
- Displaying Redundant Configurations

Overview of CSS Redundancy

Redundancy helps to ensure:

- High availability for your network applications
- Users do not experience long network delays or black holes due to a single point of failure.

A CSS provides three types of redundancy.

- Virtual IP (VIP) redundancy and virtual interface redundancy - Provide redundant VIP addresses and redundant virtual interfaces for fate sharing and server default gateways. For details, see Chapter 1, Configuring VIP and Virtual Interface Redundancy.
- Adaptive Session Redundancy (ASR) - Provides session-level redundancy (stateful failover) to continue active flows without interruption if the master CSS fails over to the backup CSS. For details, see Chapter 2, Configuring Adaptive Session Redundancy.
- Box-to-box redundancy - Provides chassis-level redundancy between two identically configured CSSs. For details, see this chapter.

The following sections provide information about when (and when not) to use the different types of redundancy.

When to Use VIP and Virtual Interface Redundancy

Typically, you configure VIP redundancy on the public side of CSS peers that are positioned in front of a server farm. You configure virtual interface redundancy on the private-side interfaces attached to the Layer 2 device in front of the servers.

Configure VIP redundancy:

- With virtual interface redundancy to provide fate sharing
- When you have a common subnet between the two CSSs on which the VIPs reside
- As a prerequisite to configuring ASR (requires active-backup VIP redundancy)
- To provide active-active CSS behavior (both CSSs processing flows)

Configure interface redundancy:

- With VIP redundancy to provide fate sharing
- When you need a default gateway for the back-end servers
- Instead of VIP redundancy on the client side of the CSS when the VIPs are on a subnet different from the subnet of your uplinks

When to Use ASR

ASR provides session-level redundancy for applications where active flows (including TCP and UDP) must continue without interruption, even if the master CSS fails over to the backup CSS.

Configure ASR:

- If you require stateful failover for mission-critical applications (for example, enterprise applications, long-lived flows, such as HTTP or FTP file transfers, and e-commerce)
- After you have first configured active-backup VIP and virtual interface redundancy

Do not configure Inter-Switch Communications (ISC) links where you have an Layer 2 device between the redundant CSS peers.

When to Use Box-to-Box Redundancy

Configure box-to-box redundancy when you:

- Expect the behavior of the CSSs to be active/standby (only the master CSS processes flows)
- Can configure a dedicated Fast Ethernet (FE) link between the CSSs for the VRRP heartbeat

Do not configure box-to-box redundancy when you:

- Expect the behavior of the CSSs to be active-active (both CSSs processing flows). Use VIP redundancy instead.
- Cannot configure a dedicated FE link between the CSSs.
- Require the connection of an Layer 2 device between the redundant CSS peers.

Redundancy Protocol Overview

The CSS redundancy protocol provides chassis-level redundancy between two CSSs. This feature protects the network and ensures that users have continuous access to servers and content.

Using the redundancy protocol CLI commands, you can configure two CSSs in a master and backup redundancy configuration. In a redundant configuration, the master CSS sends a redundancy protocol message (heartbeat) every second to inform the backup CSS that it is alive.

If the master CSS fails and does not send a message within 3 seconds, the backup CSS:

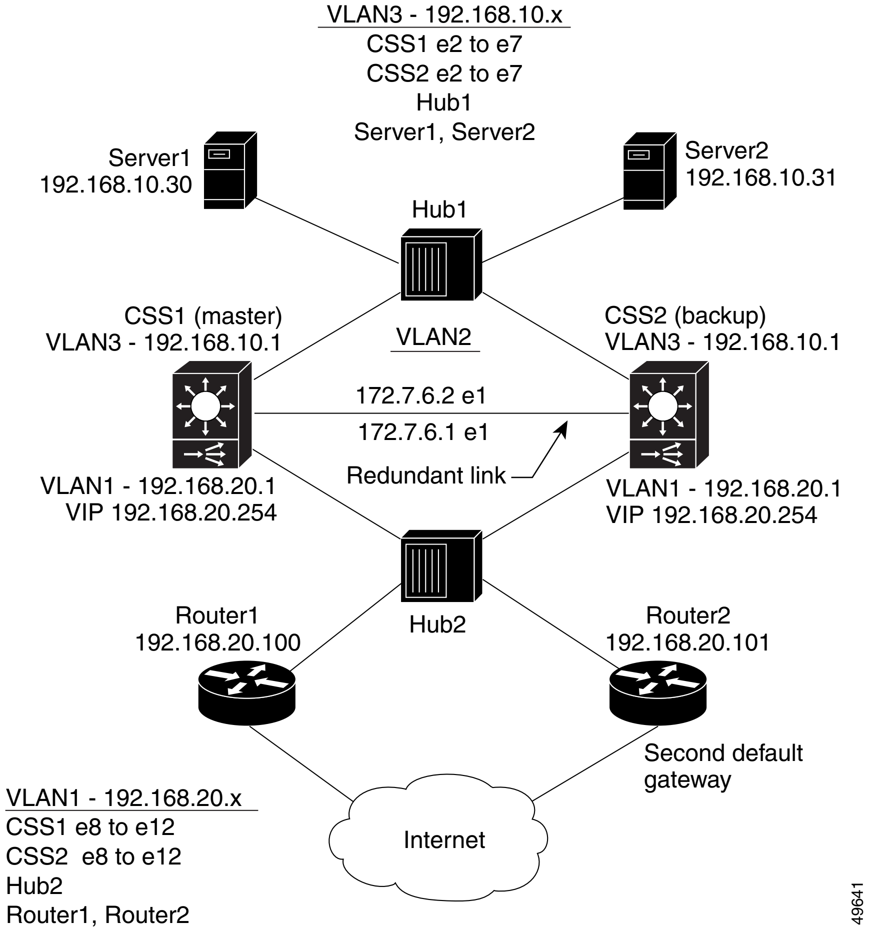
- Becomes the master CSS.
- Begins sending out redundancy protocol messages.
- Sends out gratuitous Address Resolution Protocol (ARP) messages to update the ARP tables on neighboring nodes and the forwarding tables of attached bridging devices (for example, Layer 2 switches) with the new master CSS IP address. The CSS transmits one ARP request packet and one ARP reply packet for every gratuitous ARP invocation.

If a former master comes back online, it becomes a backup CSS automatically when it receives the master CSS messages, unless you explicitly designated the CSS to be the master when you configured it. For details on IP redundancy, see "Configuring IP Redundancy" later in this chapter.

Caution When you use access control lists (ACLs) in a redundant configuration, ensure that you permit all traffic on the redundant circuit between the master and backup CSSs. For information on ACLs, refer to the *Cisco Content Services Switch Security Configuration Guide*.

Figure 3-1 shows an example of a redundant configuration with multiple VLANs. Table 3-1 uses this figure to define the command examples necessary to configure the redundancy protocol.

Figure 3-1 Redundancy Configuration Example



Redundancy Configuration Quick Start

Table 3-1 provides the steps required to configure the redundant configuration shown in Figure 3-1. Each step includes the CLI command required to complete the task. For a complete description of each feature, refer to the sections following Table 3-1.

Listed below are the basic steps to configure redundancy:

1. Install only one crossover cable on the master and redundant CSS before you power them on.

Caution If you power on the CSSs before you install the cable, both units boot up as the master CSS and cause network problems. Do not remove the crossover cable from a redundant configuration or each CSS will become master

> **Note** You must connect only one crossover cable directly to the Gigabit Ethernet (GE) ports (software version 7.10.1.02 and greater) or the Fast Ethernet (FE) ports on the redundant CSSs. Do not use Layer 2 devices between the two CSSs on the redundant link.

2. Configure each server's default gateway as the CSS's circuit VLAN IP address.
3. Configure redundancy on the existing master CSS and save the running-config to startup-config.
4. FTP the startup-config to a PC. Edit the file by including the backup CSS circuit VLAN IP addresses.
5. FTP the startup-config to the backup CSS. Reboot the backup CSS with the new startup-config.

As an alternative method, you can use CLI commands to manually configure the backup CSS with all necessary configurations including the redundancy protocol.

Note If you make configuration changes to the master CSS startup-config, you must make the same changes to the backup CSS startup-config. To learn how to synchronize the running-configs of the master CSS and the backup CSS, see "Synchronizing a Redundant Configuration" later in this chapter.

Table 3-1 Redundancy Configuration Quick Start

Task and Command Example
1. Install the crossover cable before you power up the CSSs. Make the CSS-to-CSS connection using a Category 5 crossover cable. This table uses port 1/1 on the master CSS and port 1/1 on the backup CSS as examples.
2. Configure each server's default gateway as the CSS circuit VLAN IP address.
3. Enter the **ip redundancy** command on the master CSS to enable CSS-to-CSS redundancy. `(config)# ip redundancy`
4. Configure an interface on the master CSS for a redundant connection to the backup CSS. For information on configuring interfaces, refer to the *Cisco Content Services Switch Routing and Bridging Configuration Guide*. `(config)# interface 1/1`
5. Assign the interface to the redundant connection VLAN. For information on bridging the interface to a VLAN, refer to the *Cisco Content Services Switch Routing and Bridging Configuration Guide*. `(config-if[e1])# bridge vlan 2`

6. Enter circuit mode for the redundant VLAN. For information on configuring circuits, refer to the *Cisco Content Services Switch Routing and Bridging Configuration Guide*.
`(config-if[e1])# circuit VLAN2`
`(config-circuit[VLAN2])#`

7. Assign an IP address and subnet mask to circuit VLAN2. For information on configuring a circuit IP address, refer to the *Cisco Content Services Switch Routing and Bridging Configuration Guide*. `(config-circuit[VLAN2])# ip address 172.7.6.1/24`

Table 3-1 Redundancy Configuration Quick Start (continued)

Task and Command Example

8. Enable the redundancy protocol on the redundant IP interface.
`(config-circuit-ip[VLAN2-172.7.6.1])#` **`redundancy-protocol`**
`(config-circuit-ip[VLAN2-172.7.6.1])#` **`exit`**

9. Define the other circuits as redundant circuits.
`(config-if[1/1])#` **`circuit VLAN1`**
`(config-circuit[VLAN1])#` **`redundancy`**
`(config-if[1/1])#` **`circuit VLAN3`**
`(config-circuit[VLAN3])#` **`redundancy`**

10. From SuperUser mode, save the master CSS running-config to the startup-config. `#` **`copy running-config startup-config`**
11. FTP the startup-config to a PC for editing.
12. Using a text editor, edit the startup-config by including the backup CSS circuit VLAN IP address for the redundant connection (in Figure 3-1, circuit VLAN2, IP address 172.7.6.2/24). Do not change the backup CSS VIP. The master and backup CSS must have the same VIP. If you have multiple VIPs, you must configure them on both the master and backup CSSs.
13. FTP the new file to the backup CSS and, if necessary, rename it **startup-config**.
14. Reboot the backup CSS.
15. Enter the **show redundancy** command to display the redundancy configuration and ensure that the backup CSS is configured properly.
16. Cable all hubs (or switches) to the backup CSS.

The following running-config example shows the results of entering the commands in Table 3-1.

```
!*************************** GLOBAL ***************************
  ip redundancy

!************************* INTERFACE *************************
interface  1/1
  bridge vlan 2
```

```
interface  2/1
  bridge vlan 3

!************************* CIRCUIT *************************
circuit VLAN1
  redundancy

  ip address 172.27.16.23 255.255.255.0

circuit VLAN2

  ip address 172.7.6.1 255.255.255.0
    redundancy-protocol

circuit VLAN3
  redundancy
```

Cabling Redundant CSSs

When you set up a redundant CSS configuration, install only one Category 5 crossover cable directly to the two CSSs to connect the master and backup interfaces. The CSS does not support more than one redundancy link between the two CSSs.

Caution If you power on the CSSs before you install the cable, both CSSs boot up as the master CSS and cause network problems. Do not remove the crossover cable from a redundant configuration or each CSS will become master.

Note You must connect the crossover cable directly to the Gigabit Ethernet (GE) ports (software version 7.10.1.02 and greater) or the Fast Ethernet (FE) ports before you power on the redundant CSSs. Do not use Layer 2 devices between the two CSSs on the redundant link.

Table 3-2 lists the pinouts for the CSS Fast Ethernet connectors and the crossover pinouts.

Table 3-2 RJ-45 Fast Ethernet Connector Pinouts

Signal Name	Pin Number	Crossover Cable Pinouts
RX +	1	3
RX -	2	6
TX +	3	1
Unconnected	4	4
Unconnected	5	5
TX -	6	2
Unconnected	7	7
Unconnected	8	8

Configuring Redundancy

Configuring the redundancy protocol requires you to:

- Configure the master and backup CSSs for redundancy using the **ip redundancy** command.
- Enable the redundancy protocol on the master and backup circuit VLAN using the **redundancy-protocol** command.
- Enable the circuit VLAN for redundancy using the **redundancy** command.

Note The CSS does not support IP redundancy and VIP redundancy simultaneously. For information on VIP redundancy, refer to Chapter 1, Configuring VIP and Virtual Interface Redundancy.

Configuring IP Redundancy

Use the **ip redundancy** command to enable CSS-to-CSS redundancy on two CSSs interfaced with a crossover cable. By default, redundancy is disabled on CSSs until you issue this command on both CSSs.

When you include the **master** option with this command, you can designate which CSS is the master CSS. Initially, booting two CSSs interfaced with a crossover cable determines which is the master and which is the backup. The CSS that boots first is the master CSS. If the CSSs boot at the same time, the CSS with the numerically higher IP address becomes the master.

Note You cannot use the **ip redundancy master** command with either the **type redundancy-up** command (redundancy uplink service) or the **redundancy-phy** command (physical link redundancy). If necessary, disable the appropriate command using **no type redundancy-up** or **no redundancy-phy** before using the **ip redundancy master** command.

When you issue the **ip redundancy master** command on a CSS, it becomes the master CSS. You can issue this command on either the current master or backup. If you issue this option on the backup CSS, it becomes the master and the other CSS becomes the backup automatically.

If you designate a master CSS, it regains its master status after it goes down and then comes up again. For example, when the master CSS goes down, the backup CSS becomes master. However, when the former designated master CSS comes up again, it becomes the master again.

If you have no requirement to designate a CSS as the master when both CSSs are up, do not include the **master** option when enabling redundancy on the master CSS. In this configuration, if the master CSS goes down, the backup CSS becomes master. When the former master CSS comes up again, it becomes the backup CSS.

The syntax for this global configuration mode command is:

- **ip redundancy** - Enables CSS-to-CSS redundancy on the backup CSS. If you have no requirement to define a CSS as the master CSS, use this command on both CSSs in the redundant configuration.

- **ip redundancy master** - Enables CSS-to-CSS redundancy on the CSS that you want to designate as the master CSS. (Be sure to issue **ip redundancy** on the backup CSS.) You can issue **ip redundancy master** on a CSS:
  - Whether or not it was initially booted as the master or the backup. If you issue this command on the backup CSS, it becomes the master and the other CSS becomes the backup CSS automatically.
  - When CSS-to-CSS redundancy is currently enabled.

For example:

```
(config)# ip redundancy
```

Caution You cannot issue the **ip redundancy master** command on both the master and backup CSSs. This can cause severe network problems. Before you disable redundancy, ensure that you disconnect or disable all redundant circuits to prevent duplicate IP addresses from occurring on the network.

To disable CSS-to-CSS redundancy, enter:

```
(config)# no ip redundancy
```

Note The **no ip redundancy** command deletes the **redundancy** and **redundancy-protocol** commands from the running-config of the CSS.

Before the CSS disables redundancy, it displays the following message:

```
WARNING: Disabling redundancy may result in duplicate
IP addresses on the network.
Be sure you disconnect or disable all redundant circuits before you
disable redundancy.
Do you want to disable redundancy? [y/n]:
```

Type **y** to disable redundancy. Type **n** to cancel disabling redundancy.

To unassign the CSS as the master CSS, enter:

```
(config)# no ip redundancy master
```

Note The **no ip redundancy master** command does not disable CSS-to-CSS redundancy.

Configuring Redundant Circuits

To configure a circuit as a redundant circuit, use the **redundancy** command. The **redundancy** command is available in circuit configuration mode.

Note The redundancy command causes the specified VLAN to become silent when in backup mode.

When you configure redundancy, configure it on circuits (VLANs) that contain network IP addresses shared by the redundant CSSs (in Figure 3-1, these are VLAN1 and VLAN3). Do not configure redundancy on the circuit (VLAN) configured specifically for redundancy communications (in Figure 3-1, this is VLAN2).

The example below configures VLAN1 as a redundant circuit, which contains a shared network IP address (in Figure 3-1, this is 192.168.20.1).

For example:

```
(config-circuit[VLAN1])# redundancy
```

To remove a circuit from the redundancy configuration, enter:

```
(config-circuit[VLAN1])# no redundancy
```

Configuring the Redundancy Protocol

To configure the redundancy protocol on the circuit (VLAN) connecting the two CSSs, enter the **redundancy-protocol** command in IP interface configuration mode. Enter the command for the circuit you configured specifically for redundancy communications (in Figure 3-1, this is VLAN2).

Note The CSS box-to-box redundancy protocol is supported on CSS Gigabit Ethernet (GE) ports in software version 7.10.1.02 and greater.

For example:

```
(config-circuit-ip[VLAN2-172.7.6.1])# redundancy-protocol
```

To stop running a redundancy protocol on an interface, enter:

```
(config-circuit-ip[VLAN2-172.7.6.1])# no redundancy-protocol
```

Configuring the VRRP Backup Timer

Configure the **vrrp-backup-timer** command on both redundant CSSs to specify the time interval in seconds that the backup CSS waits to assume mastership when the master CSS goes down. Because timer values greater than the 3-second default cause longer failover times, use this command only in environments where the CPU utilization on the CSS is close to 100 percent. After you set the timer value, you need to reissue the **redundancy-protocol** command on the redundant circuit between the CSSs for the new timer value to take effect. For details on configuring the redundancy protocol, see "Configuring the Redundancy Protocol" earlier in this chapter.

Note If you intend to use the commit_redundancy script to synchronize your redundant configuration, be sure to specify the -a argument in the **script play** command to ensure that the script copies the timer configuration setting from the master CSS to the backup CSS. For details on synchronizing your redundant configuration, see "Synchronizing a Redundant Configuration" later in this chapter.

The syntax for this global configuration mode command is:

vrrp-backup-timer *wait_time*

The variable for this command is *wait_time*. Enter an integer from 3 to 120 seconds. The default is 3 seconds.

For example:

```
(config)# vrrp-backup-timer 15
```

To reset the timer to the default value of 3 seconds, enter:

```
(config)# no vrrp-backup-timer
```

Synchronizing a Redundant Configuration

To ensure that your backup CSS can perform the same tasks as your master CSS in the event of a master CSS failure, the running-config on the backup must be identical to the running-config on the master. To automate this configuration synchronization process, you can run a script (commit_redundancy) on the master CSS that copies the master CSS running-config to the backup CSS running-config.

There are two types of configuration synchronization:

- **Complete** - On CSSs that have an identical chassis (the same CSS model), produces a running-config on the backup CSS that exactly matches the running-config on the master CSS.
- **Partial** (default) - On CSSs with an incompatible configuration syntax, synchronizes all parameter values in the configuration except the interface and circuit configurations. For example, the master is a CSS 11506 and the backup is a CSS 11501. The script maintains the current backup interface and circuit configurations automatically.

Before You Begin

Before you run the configuration synchronization script, ensure that you have set up the redundancy circuit between the two CSSs and that the Application Peering Protocol (APP) is running on that circuit. For details on configuring the redundancy circuit, see “Configuring Redundancy” earlier in this chapter. For details on configuring APP, refer to the *Cisco Content Services Switch Global Server Load-Balancing Configuration Guide*.

If you configure the **restrict user-database** command on a CSS, only users with administrative or technician privileges can configure the **username** command. To be consistent with the **restrict user-database** command, the **commit_redundancy** script does not modify the privilege level of the administrative or technician users on the backup CSS.

To run **commit_redundancy** successfully users with administrative and technician privileges must be identical on both CSSs. You cannot have a local user (configured with the **username** command) on the master CSS and an administrative or technician user with the same username on the backup CSS.

For more information about the **restrict user-database** command, refer to the *Cisco Content Services Switch Security Configuration Guide*. For more information about configuring administrators (**username_offdm** command) and technicians (**username_technician** command), refer to the *Cisco Content Services Switch Command Reference*.

Running the Configuration Synchronization Script

To run the configuration synchronization script, use the **script play commit_redundancy** command in SuperUser mode. The syntax is:

script play commit_redundancy "*arguments***"**

You can also run the configuration synchronization script using the predefined alias that comes with all CSSs by entering:

```
# commit_RedundConfig "arguments"
```

The arguments for the commit_redundancy script are:

- *ip address* - The IP address of the backup CSS. This is the only required argument for this script. For details on automating the entry of the IP address, see "Setting the REMOTE_IP Variable" later in this section.
- **-a** (All) - Performs a complete configuration synchronization. Use this option only when the master CSS and the backup CSS have identical chassis (see Table 3-3).
- **-d** (Debug) - Debug switch for the commit_redundancy script, which displays the current task being performed as the script progresses. Debug messages display even when you specify the **-s** argument.

Caution Before you use the **-f** argument to remove a config sync lock file, ensure that no one else is running the config sync script on the CSS. Otherwise, if you remove the lock file and then run the script again while the script is in use, the resulting configurations may have some discrepancies.

- **-f** - After an abnormal script termination, removes the lock file so that you can run the script again. This argument overrides all other specified arguments and the script exits immediately after removing the lock file. For details on the lock file, see "Setting the REMOTE_IP Variable" later in this section.
- **-int** (Interface) - Does not clear the interfaces on the backup CSS so that the link does not go down. Do *not* use this argument with the **-a** argument. If you do and the interface settings are different on the master and the backup CSSs, the configurations will not match and the script will not finish successfully.
- **-nv** (No Verify) - Informs the script not to verify that the configuration synchronization was successful. The script does inform you if the script fails.

  Note The script verifies the configuration synchronization by default.

- **-s** (Silent) - Suppresses script progress messages and displays only the result of running the script: Config Sync Successful or Config Sync Failed.

  Note The -s option is not compatible with running the commit_redundancy script on the backup CSS. If you run the script on the backup CSS using the -s option, the script exits and no configuration changes are made.

Note You can specify the script arguments in any order.

For example:

```
CSS11503# script play commit_redundancy "10.7.100.93 -d -s"
```

The following output appears:

```
Verifying that IP redundancy is activated on Master switch.

Verifying that app session is up with backup switch.

Making sure app session is up.

Checking Master redundancy-config for redundancy-protocol set and if
so storing it in variable MASTER_IP.
```

```
Verify that the IP Address specified is the Backup IP Address.

Making sure app session is up.

Saving Master running-config to startup-config and archiving
startup-config.
Copying running-config to startup-config.

Archiving startup-config.

Copying startup-config to a temp file tmp.cfg.
Swapping Master and Backup ip addresses in tmp.cfg for app

Removing CIRCUIT and INTERFACE modes from tmp.cfg.

Checking if IP redundancy master is set.

Using rcmd to copy tmp.cfg to a file on Backup switch.

Retrieving circuit info for redundancy-protocol link.

Archiving copy to Backup startup-config.

Archiving Backup current startup-config.

Restoring startup-config (new copy) to startup-config.

Clearing running-config.

Script playing the copy script of the Master running-config.

Making sure app session is down.

Copy success being verified by comparing byte sizes of archived
running-configs of the Master switch and the Backup switch.

Making sure app session is up.

Comparing the byte count now.

Config Sync Successful.
```

In this example, the script:

- Performs a partial configuration synchronization (default)
- Displays the current task being performed as the script progresses (-d)
- Suppresses progress messages (-s)
- Verifies that the configuration synchronization was successful (-v)

For more information on scripts, refer to the *Cisco Content Services Switch Administration Guide*.

Config Sync Lock File

When you run the script, the software creates a lock file (config_sync_lock) in the script directory so that you cannot run the script from another session to the CSS. If the lock file exists and you run the script, the following message appears:

```
The script is in use by another session.
```

If the script terminates abnormally, the software does not remove the lock file. The next time you run the script, the above message appears. If you are certain that the script is not in use by another session, then you can use the -f argument to remove the lock file.

When you run the script with the -f argument, the following message appears and the script exits:

```
Config Sync lock file removed.
```

Now you can run the script again.

Setting the REMOTE_IP Variable

To eliminate the need to specify a remote IP address each time you run the commit_redundancy configuration synchronization script, you can set the value of the variable REMOTE_IP to an IP address and save it in your user profile. Once you set the variable and save it in your user profile, the variable will always be available after you log in to the CSS.

Set the REMOTE_IP variable to the APP session IP address configured on the local CSS. The APP session address is the circuit IP address for the remote CSS. To set the variable, enter:

```
# set REMOTE_IP "remote_ip_address" session
```

To save the variable in your user profile, enter:

```
# copy profile user-profile
```

If you already created the BACKUP_IP variable in an earlier release, the script will use the new variable instead, if present.

Logging Configuration Synchronization Script Result Messages

You can specify that script result messages (script success or failure messages) be sent to the current logging device automatically each time you run the configuration synchronization script. To log the script result messages, enable logging on NETMAN with level info-6 or debug-7 by entering:

```
(config)# logging subsystem netman level info-6
```

Note Log messages are generated with or without the -s (silent) argument specified. See "Running the Configuration Synchronization Script" earlier in this chapter.

For example, if the APP session to the backup CSS is not running, the following log message will be generated:

```
config sync: app session is DOWN
```

For ease of tracking, each log message contains the string "config sync".

Using the Redundancy Force-Master Command

Use the **redundancy force-master** command to configure a backup CSS as a master *temporarily*. This is a temporary setting because it is not copied to the running-config. This command is useful in a redundant configuration when you need to take the master CSS offline for maintenance or an upgrade.

By issuing the **redundancy force-master** command on the backup CSS in global configuration mode, you set that CSS to master and ensure that users have continuous access to servers and content. The forced master CSS remains the master:

- Until it goes down and comes back up as the backup, or
- You manually make the other CSS the master, using either the **redundancy force-master** command or the **ip redundancy master** command.

Note If you explicitly designated the master CSS using the **ip redundancy master** command, you cannot use the **redundancy force-master** command on the backup CSS. In this case, you must unassign the master CSS by issuing the **no ip redundancy master** command before you can use the **redundancy force-master** command on the backup CSS.

Configuring Multiple Redundant Uplink Services

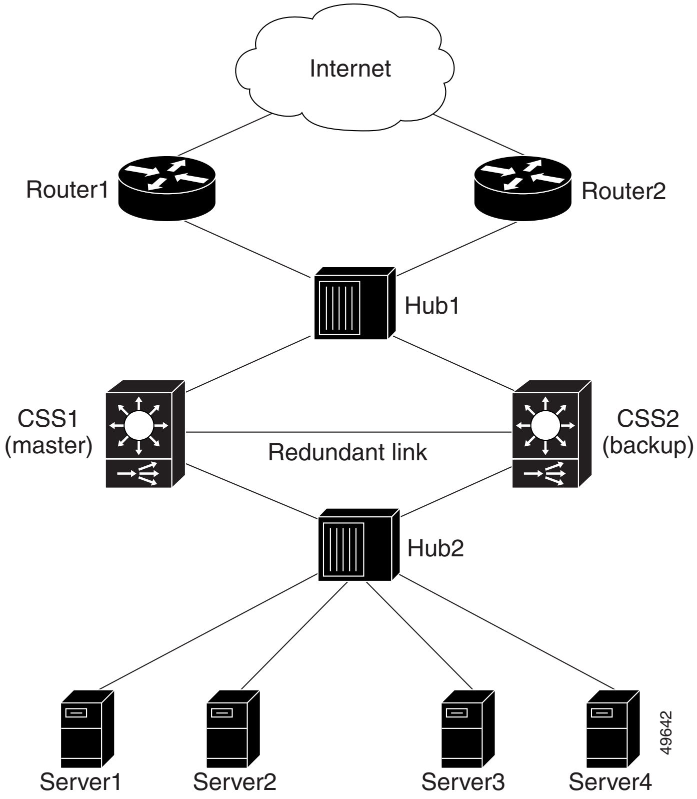
Within a redundant configuration, the CSS allows you to configure multiple redundancy uplink critical services (up to a maximum of 512). Use the **type redundancy-up** command to designate one or more routers as type redundancy-up critical services. (A typical configuration contains 10 or fewer routers.) This critical service type enables the master CSS to ping a router service using the default keepalive Internet Control Message Protocol (ICMP). If the master CSS fails or it detects that all router uplink critical services have failed, the backup CSS becomes the master.

In a redundant configuration that does not configure the routers as type redundancy-up critical services, a backup CSS becomes master only when the current master CSS fails. In this configuration, a switchover *does not* occur when the router services fail.

Note You cannot add redundancy uplink critical services to a content rule.

Figure 3-2 shows a typical redundant configuration. When CSS1 fails or CSS1 cannot communicate with both the Router1 critical service and the Router2 critical service, CSS2 becomes the master CSS automatically.

Figure 3-2 Multiple Redundant Uplink Services Configuration Example



Note If you explicitly designated the master CSS using the **ip redundancy master** command, you cannot use the **type redundancy-up** command on the CSS. In this case, you must unassign the master CSS by issuing the **no ip redundancy master** command before you can use the **type redundancy-up** command.

Use the **type redundancy-up** command to configure each router service as a redundancy uplink critical service. For example:

```
(config-service[router1])# type redundancy-up
(config-service[router1])# ip address 192.168.1.1
(config-service[router1])# active
```

Use the **show redundancy** command to display critical services. See the "Displaying Redundant Configurations" section.

For example:

```
CSS1(config)# show redundancy
```

Using the redundancy-phy Command

Use the **redundancy-phy** command in interface mode to add an interface to the physical link configuration list. If any physical link in the configuration list goes down, the master CSS fails over to the backup CSS. You can configure a maximum of 32 interfaces. The CSS saves this configuration information to the running-config.

For example:

```
(config-if[2/1])# redundancy-phy
```

Note If you configure the **redundancy-phy** command on an interface of the master CSS and then make any change to the port settings of that interface using the **phy** command (for example, changing **auto-negotiate** to **100Mbits-FD**), the master CSS fails over to the backup CSS. To prevent the failover from occurring, enter the **no redundancy-phy** command on the interface first, change the port settings, and then reenter the **redundancy-phy** command.

You cannot use the **redundancy-phy** command if you used the **ip redundancy master** command to configure the master CSS. In this case, you must enter the **no ip redundancy master** command before you can use the **redundancy-phy** command.

To disable a configured interface and delete it from the physical link list, enter:

```
(config-if)# no redundancy-phy
```

Note If you configure the **redundancy-phy** command on an interface and then disable the interface using the **admin-shutdown** command, the master CSS fails over to the backup CSS. To prevent the CSS from failing over when you administratively disable the interface, remove the **redundancy-phy** command by entering **no redundancy-phy** before you enter the **admin-shutdown** command on that interface.

When you use the **redundancy-phy** command and both CSSs are connected to a Layer 2 switch, be sure to monitor physical link failure only on the critical physical links and not on the redundant link between the two CSSs. This will avoid the detection of a physical link down and possible thrashing when one of the CSSs is rebooting or transitioning between master and backup states.

Configuring Stateless Redundancy Failover

Use the **redundancy-l4-stateless** command in content or group configuration mode to enable stateless redundancy failover in a box-to-box redundancy or a VIP and virtual interface redundancy configuration. Stateless redundancy failover allows critical TCP/IP traffic to continue in case of a failure at the load-balancing CSS by allowing the backup CSS to set up a mid-stream TCP flow. This feature is disabled by default.

The default behavior of a CSS is to set up load-balanced TCP flows only when it receives a TCP frame that begins with SYN. When stateless redundancy failover is enabled and a failover occurs, the backup CSS establishes a mid-stream flow for any existing TCP sessions. The CSS still exhibits the default behavior for all new flows. To restore the default behavior of the CSS for all flows after issuing the **redundancy-l4-stateless** command, use the **no redundancy-l4-stateless** command.

Note This feature affects only TCP/IP sessions. UDP behaves normally because UDP is not a a session-oriented protocol.

Before You Begin

Before you attempt to implement stateless redundancy failover for the first time, read this section in its entirety. You should already be familiar with the following concepts:

- Redundancy
- Layer 3 and layer 4 content rules
- Virtual IP addresses (VIPs)
- Load balancing
- Source groups
- Services
- Keepalives
- Convergence

Environmental Considerations

Stateless redundancy failover requires a very specific redundant CSS configuration, where the state of the CSS can be determined after a failure. This feature supports redundant routes in the high-availability topology surrounding the CSSs. However, the topology must *not* balance packets in a TCP/IP socket connection across more than one Ethernet port on the CSS.

Routed paths to the load-balanced VIP should be weighted to ensure that a single path is preferred for the lifetime of a TCP/IP connection.

Note Stateless redundancy failover does not support network address translation (NAT) where maintaining state is required nor does it support Layer 5 content rules.

General Configuration Requirements

The following sections describe the stateless failover requirements that apply to both box-to-box redundancy and VIP and virtual interface redundancy configurations.

Configuration Restrictions

The following configuration restrictions apply to all CSSs, except where noted.

- Stateless redundancy failover is incompatible with service remapping (the **persistence reset remap** command). Stateless redundancy failover requires that the CSS not NAT client source ports. Backend remapping enables CSS port mapping, which NATs source ports for all flows. For more information on service remapping, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.
- Configuring session-level redundancy and stateless redundancy failover on the same CSS is not supported.
- Do not configure a service that changes the destination port on a content rule. This causes the CSS to NAT (port map) the destination port. If the CSS fails over, the backup CSS has no knowledge of the original destination port.

Configuring CSS Parameters

The following parameters must match exactly on both redundant CSSs:

- **Stateless redundancy failover command -** Include the **redundancy-l4-stateless** command in both the content rules and the source groups associated with the redundant VIP.
- **Content rules** - Create identical content rules on both CSSs with the following parameters. Refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.
  - **VIP** - Assign a virtual IP address to each content rule. No wildcard addresses are allowed and no VIP ranges on the content rule are allowed.

  - **Load-balancing method** - Configure the load-balancing method as source IP address (srcip), the only load-balancing method that is supported by stateless redundancy failover.
  - **Failover method** - Configure either 'linear' (default) or 'next' type as the service failover method. 'Bypass' is not supported.
- **Services** - For each load-balanced server farm, configure the following service-related parameters to be the same on both CSSs for each content rule. Refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.
  - **Service name** - Use identical service names on the master and the backup CSS. Service names are case-sensitive.
  - **IP address** - Use identical IP addresses on the master and the backup CSS.

Note Configured services may not change the CSS destination port. In a stateless environment, there is no way to determine the original destination port when the packet returns from the server.

  - **Service number and order** - The CSS orders services internally in alphabetical order regardless of the order in which you enter them in the configuration.
  - **Keepalives** - Create keepalives using the global **keepalive** command, then associate the services with the keepalives using the **keepalive type named** command. Both CSSs must be able to send and receive keepalive messages with the same servers. This helps to ensure that a redistribution of the balance method does not occur in a failover event.
  - **Weight** - Routed paths to the load-balanced VIP should be weighted to ensure that a single path is preferred for the lifetime of a TCP/IP connection.
- **Source groups** - Create a source group with the same VIP as the content rule VIP on each CSS to NAT source addresses for packets returning from the server. In case of a failover, the source group will handle the connection setup for TCP/IP transmissions that arrive at the CSS from the servers. All servers in the farm must be members of the configured source group. Refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

Note Do not configure source groups for outbound traffic from the servers, because the backup CSS does not know which ports were NATed by the source group on the master CSS if a failure occurs at the master CSS. This restriction also applies to active FTP because the server initiates the data connection.

Synchronizing the CSS Configurations

You can manually synchronize the CSS configurations by ensuring that the configurations are exactly the same. In an IP redundancy configuration, you can run the commit_redundancy configuration synchronization script. The script automatically synchronizes the master and backup CSS configurations. See "Synchronizing a Redundant Configuration" earlier in this chapter.

Box-to-Box Redundancy Configuration

For details on box-to-box redundancy, see the earlier sections in this chapter.

In case of a failure on the master CSS, the backup CSS becomes the master CSS. The following actions occur:

1. All VLANS become active.
2. Topology protocols (for example, Spanning Tree) initialize and converge.
3. All configured interface (circuit) and VIP addresses are acquired by gratuitous ARP.
4. The master CSS acquires servers through keepalives.

Note If your configuration is large or the servers respond slowly, the completion of step 4 may take several seconds.

Complex topologies surrounding the CSS converge after the CSS has determined a root bridge and has begun transmission of Address Resolution Protocol (ARP) and keepalive traffic. If a TCP/IP retransmission from a server arrives at the CSS before the CSS acquires the server, the CSS sets up the connection properly through the configured source group path. If a retransmission from a client arrives at the CSS before all servers have been acquired and the source IP address of the client indicates a server that is not yet alive, the CSS sets up the connection according to the failover method configured in the content rule (next or linear).

Layer 2 and Layer 3 Configuration and Convergence

Because IP Redundancy disables the forwarding of traffic through VLANs on the backup CSS, configure the CSS to provide either a bridged or routed path between servers and uplink routers. In either case, the CSS must be the default gateway for load-balanced servers. Refer to the *Cisco Content Services Switch Routing and Bridging Configuration Guide*.

If you configure the CSSs with servers and a balance VIP on the same VLANs as the uplink router (bridged mode), then configure the CSSs to not send ICMP redirects to the servers, using the **no redirect** command. Refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

Configuration Example

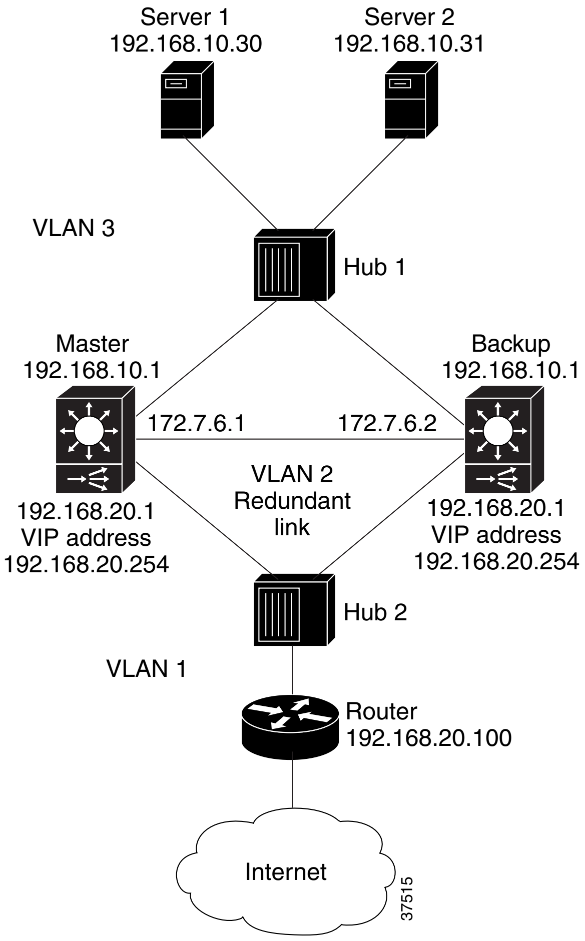
The following example configuration (see Figure 3-3) assumes that the:

- CSSs are acting as routers between two external VLANs in an IP Redundancy configuration.
- External VLANS exist on Layer 2 and Layer 3 devices.
- Layer 2 and Layer 3 devices are not the point of failure.

```
ip route 0.0.0.0.0.0.0.0.192.168.20.100
ip redundancy

interface e2
    bridge vlan 1
    description “uplink VLAN”

interface e5
    bridge vlan 1
    description “uplink VLAN”
```

```
interface e9
    bridge vlan 3
    description “server VLAN”

interface e12
    bridge vlan 3
    description “server VLAN”

interface e1
    bridge vlan 2
    description “Redundancy Protocol Heartbeat”
circuit VLAN1
    redundancy
    ip address 192.168.20.1 255.255.255.0

circuit VLAN3
    redundancy
    ip address 192.168.10.1 255.0.0.0

circuit VLAN2
    redundancy-protocol
    ip address 172.7.6.1 255.255.255.253

service s1
    ip address 192.168.10.30
    active
service s2
    ip address 192.168.10.31
    active

owner Redundant-Pool
    content web
        vip address 192.168.20.254
        protocol tcp
        port 80
        redundancy-l4-stateless
        add s1
        add s2
        balance srcip
        active

group Redundant-Pool
    vip address 192.168.20.254
    redundancy-l4-stateless
    add service s1
    add service s2
    active
```

Figure 3-3 Example Box-to-Box Redundancy Configuration for Stateless Redundancy Failover



VIP and Interface Redundancy Configuration

For details on VIP and virtual interface redundancy, refer to Chapter 1, Configuring VIP and Virtual Interface Redundancy.

Layer 2 and Layer 3 Configuration and Convergence

A CSS that is running Virtual Router Redundancy Protocol (VRRP) does not shut down any VLANs. Therefore, VRRP configurations may not be configured with content rule VIP, uplink, and server addresses in the same VLAN (bridged mode). Instead, the CSSs must be configured so that both CSSs in a redundant pair act as routers between the uplink VLAN and the server VLAN. The CSS uses a virtual router address for the default gateway on the servers.

Because both CSSs are active and participating in topology protocols, convergence time may be reduced in the event of a failure. Additionally, both CSSs acquire servers with keepalive traffic at all times, so that both CSSs agree on server availability.

VRRP provides for a redundant routed path out of a VLAN, but does not address synchronization of more than one VLAN in the decision. Due to this limitation, ensure that one CSS does not become master for the connection to the uplink VLAN, while the other CSS is master for the connection to the server VLAN. To avoid this split state, a CSS can monitor critical external IP addresses as part of the extended VRRP implementation.

Typically, you configure a single CSS with the highest priority and the **preempt** option for each VRID pair (uplink VLAN side/server VLAN side). This ensures that if the designated CSS is available, both VRIDs will converge there, avoiding a split state.

To address more complex failure scenarios, use a script keepalive. For details on script keepalives, refer to the *Cisco Content Services Switch Administration Guide*.

In the example that follows, the master CSS relinquishes control of both virtual interfaces upon loss of contact with either the uplink router or all web servers.

Configuration Example

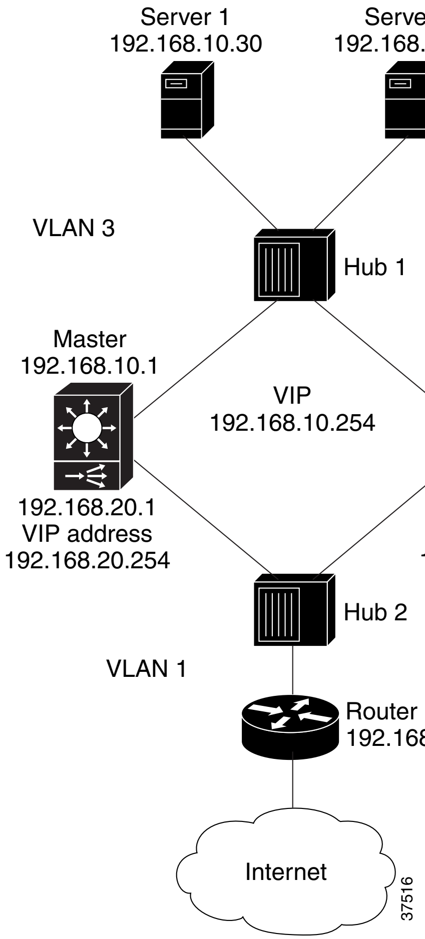
The following example configuration (see Figure 3-4) assumes that the:

- CSSs are acting as routers between two external VLANs in a VIP and virtual interface redundancy configuration.
- External VLANS exist on Layer 2 and Layer 3 devices.
- Layer 2 and Layer 3 devices are not the point of failure.

```
ip route 0.0.0.0 0.0.0.0 192.168.20.100

interface e2
    bridge vlan 1
    description “uplink VLAN”

interface e5
    bridge vlan 1
    description “uplink VLAN”

interface e9
    bridge vlan 3
    description “server VLAN”

interface e12
    bridge vlan 3
    description “server VLAN”

circuit VLAN1
    ip address 192.168.20.1 255.255.255.0
        ip virtual-router 1 priority 110 preempt
        ip redundant-vip 1 192.168.20.254
        ip redundant-interface 1 192.168.20.2
        ip critical-service 1 uplink
        ip critical-service 1 s1
        ip critical-service 1 s2

circuit VLAN3
    ip address 192.168.10.1.0.0.0
        ip virtual-router 1 priority 110 preempt
        ip redundant-vip 1 192.168.10.254
        ip redundant-interface 1 192.168.10.2
        ip critical-service 1 uplink
        ip critical-service 1 s1
        ip critical-service 1 s2
```

```
service uplink
    ip address 192.168.20.100
    type redundancy-up
    active

service s1
    ip address 192.168.10.30
    active
service s2
    ip address 192.168.10.31
    active

owner Redundant-Pool
    content web
        vip address 192.168.20.254
        protocol tcp
        port 80
        redundancy-l4-stateless
        add s1
        add s2
        balance srcip
        active

group Redundant-Pool
    vip address 192.168.20.254
    redundancy-l4-stateless
    add service s1
    add service s2
    active
```

Figure 3-4 Example of VIP and Virtual Interface Redundancy Configuration for Stateless Redundancy Failover



Alternative Configurations

Stateless redundancy failover allows other possible configurations and topologies. To use this feature in other high-availability environments, see the other sections in this chapter and to Chapter 1, Configuring VIP and Virtual Interface Redundancy, for details and examples of CSS redundancy configurations. Refer to RFC-2338 *Virtual Router Redundancy Protocol* for additional information.

Managing Your Configuration

If you need to take a server offline for maintenance, you should also take the corresponding server offline at the redundant CSS. Failing to synchronize the state of the server farms results in mismapped connections if a failover occurs during the maintenance period. You can synchronize the service states in an IP redundancy configuration automatically by running the configuration synchronization script (commit_redundancy). For a VIP/interface redundancy configuration, manually synchronize the service states.

Other Considerations

The following conditions apply to stateless redundancy failover:

- After a failover, passive mode FTP will not continue because the NAT state of the data channel cannot be preserved. However, port mode FTP will continue to function.
- Because the port-map function of source groups is disabled, connections originated by the servers in the redundantly balanced farm do not have the benefit of source port translation. This may affect functions such as DNS.
- Service records may not be configured to change the destination port of traffic that is balanced.

- At any given time, some TCP/IP connections may be either in a state where the client is sending data to the server, or the server is sending data to the client. Packets that arrive while the topology is converging or before some services are acquired by keepalive traffic may be forwarded incorrectly. For example, a service may stop responding to keepalive traffic, but continue to service a long-lived TCP connection. In this case, the backup CSS would not have knowledge of the state of the long-lived connection, and would guess incorrectly when attempting to resume the connection.
- In a highly critical environment, set goals for connection loss ratio and convergence time. Then, test various topologies and topology protocol combinations to verify that the target connection loss ratios and convergence time goals are reached. This testing should account for all reasonable failure modes that the high-availability network is designed to withstand. If warranted by the critical nature of the traffic, you may want to construct a permanent testbed to validate the system of network components prior to the deployment of new configurations.

Displaying Redundant Configurations

To display CSS-to-CSS redundancy, use the **show redundancy** command.

For example:

```
(config)# show redundancy
```

When redundancy is not configured, the CSS displays the following status:

```
(config)# show redundancy
Redundancy: Disabled Redundancy Protocol: Not Running
```

The output of the **show redundancy** command varies depending on whether you issue the command on the master or the backup CSS.

Table 3-3 describes the fields in the **show redundancy** output.

Table 3-3 Field Descriptions for the show redundancy Command

Field	Description
Redundancy	Indicates whether or not redundancy is enabled on the CSS.
Redundancy Protocol	Indicates whether or not the redundancy protocol is running on the CSS.
Redundancy State	The current redundancy state of the CSS (Master or Backup).
MasterMode	Indicates whether the CSS is configured as master. Yes indicates that the CSS is designated as master through the **ip redundancy master** command. No indicates that the CSS is not the designated master through the **ip redundancy** command.
Number of times redundancy state changed to Master/Backup	The number of times that the CSS has changed to master or backup.
Redundancy interface	The address for the redundancy interface.
Current State Duration	How long the CSS has been in its current redundancy state (master or backup).
Last Fail Reason	A description of the last CSS redundancy failure.
VRID	The virtual router identifier (VRID).
Priority	The priority for the virtual router with its peer. The default priority value is 100. Enter an integer between 1 and 255.
Physical Link Failure Monitor on	
Interface/State	The list of interfaces configured through the **redundancy-phy** command and their states. The **show** output is sorted numerically by interface port number.
Uplink Enabled	The number of enabled service uplinks.

Table 3-3 Field Descriptions for the show redundancy Command (continued)

Field	Description
Number Alive	The number of alive (Up state) service uplinks.
Service Name/State	The list of uplink services and their states. The **show** output is sorted numerically by service index number.

INDEX