CISCO SYSTEMS

Cisco Content Services Switch Routing and Bridging Configuration Guide

Software Version 8.20
November 2006

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-8241-02

CONTENTS

Preface

This guide provides instructions to configure interfaces and circuits, spanning-tree bridging, Open Shortest Path First (OSPF), Address Resolution Protocol (ARP), Routing Information Protocol (RIP), Internet Protocol (IP) routing, and Dynamic Host Configuration Protocol (DHCP). Information in this chapter applies to all 11500 Series Content Services Switch (CSS) models, except where noted.

This preface contains the following major sections:

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 Interfaces and Circuits	Configure the CSS interface ports and circuits for operation.
Chapter 2, Configuring Spanning-Tree Bridging for the CSS	Configure spanning-tree bridging.
Chapter 3, Configuring Open Shortest Path First	Configure OSPF routing protocol.
Chapter 4, Configuring the Address Resolution Protocol	Configure Address Resolution Protocol (ARP).
Chapter 5, Configuring Routing Information Protocol	Configure Routing Information Protocol (RIP).
Chapter 6, Configuring the Internet Protocol	Configure Internet Protocol (IP) routing.

Chapter	Description
Chapter 7, Configuring the Cisco Discovery Protocol	Configure Cisco Discovery Protocol (CDP).
Chapter 8, Configuring the DHCP Relay Agent	Configure Dynamic Host Configuration Protocol (DHCP).

Related Documentation

In addition to this document, the CSS documentation set includes the following:

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.

Document Title	Description
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 on 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
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

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
Cisco Content Services Switch Redundancy Configuration Guide	This guide describes how to perform CSS redundancy configuration tasks, including: • VIP and virtual interface redundancy • Adaptive session redundancy • Box-to-box redundancy

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 • Back-end 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.

Italic 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. This section explains the product documentation resources that Cisco offers.

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

The Product Documentation DVD is a library of technical product documentation on a portable medium. The DVD enables you to access installation, configuration, and command guides for Cisco hardware and software products. With the DVD, you have access to the HTML documentation and some of the PDF files found on the Cisco website at this URL:

http://www.cisco.com/univercd/home/home.htm

The Product Documentation DVD is created and released regularly. DVDs are available singly or by subscription. Registered Cisco.com users can order a Product Documentation DVD (product number DOC-DOCDVD= or DOC-DOCDVD=SUB) from Cisco Marketplace at the Product Documentation Store at this URL:

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

Ordering Documentation

You must be a registered Cisco.com user to access Cisco Marketplace. Registered users may order Cisco documentation at the Product Documentation Store at this URL:

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

If you do not have a user ID or password, you can register at this URL:

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

Documentation Feedback

You can provide feedback about Cisco technical documentation on the Cisco Technical Support & Documentation site area by entering your comments in the feedback form available in every online document.

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 will find information about how to do the following:

- 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, security notices, and security responses for Cisco products is available at this URL:

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

To see security advisories, security notices, and security responses as they are updated in real time, you can subscribe to the Product Security Incident Response Team Really Simple Syndication (PSIRT RSS) feed. Information about how to subscribe to the PSIRT RSS feed is found at 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 have identified a vulnerability in a Cisco product, contact PSIRT:

- For emergencies only—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.

- For 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 (for example, GnuPG) to encrypt any sensitive information that you send to Cisco. PSIRT can work with information that has been encrypted with PGP versions 2.*x* through 9.*x*.

Never use a revoked encryption key 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.

If you do not have or use PGP, contact PSIRT to find other means of encrypting the data before sending any sensitive material.

Product Alerts and Field Notices

Modifications to or updates about Cisco products are announced in Cisco Product Alerts and Cisco Field Notices. You can receive Cisco Product Alerts and Cisco Field Notices by using the Product Alert Tool on Cisco.com. This tool enables you to create a profile and choose those products for which you want to receive information.

To access the Product Alert Tool, you must be a registered Cisco.com user. (To register as a Cisco.com user, go to this URL: http://tools.cisco.com/RPF/register/register.do) Registered users can access the tool at this URL: http://tools.cisco.com/Support/PAT/do/ViewMyProfiles.do?local=en

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 Tool** to locate your product serial number before submitting a request for service online or by phone. You can access this tool from the Cisco Technical Support & Documentation website by clicking the **Tools & Resources** link, clicking the **All Tools (A-Z)** tab, and then choosing **Cisco Product Identification Tool** from the alphabetical list. This 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.

Tip Displaying and Searching on Cisco.com

If you suspect that the browser is not refreshing a web page, force the browser to update the web page by holding down the Ctrl key while pressing F5.

To find technical information, narrow your search to look in technical documentation, not the entire Cisco.com website. On the Cisco.com home page, click the **Advanced Search** link under the Search box and then click the

Technical Support & Documentation radio button.

To provide feedback about the Cisco.com website or a particular technical document, click **Contacts & Feedback** at the top of any Cisco.com web page.

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)—An existing 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 operations 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 the network is impaired while 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.

- The Cisco Online Subscription Center is the website where you can sign up for a variety of Cisco e-mail newsletters and other communications. Create a profile and then select the subscriptions that you would like to receive. To visit the Cisco Online Subscription Center, go to this URL:

  http://www.cisco.com/offer/subscribe

- The *Cisco Product Quick Reference Guide* is a handy, compact reference tool that includes brief product overviews, key features, sample part numbers, and abbreviated technical specifications for many Cisco products that are sold through channel partners. It is updated twice a year and includes the latest Cisco channel product offerings. To order and find out more about the *Cisco Product Quick Reference Guide*, go to this URL:

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

- 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

- *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 where networking professionals 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

- "What's New in Cisco Documentation" is an online publication that provides information about the latest documentation releases for Cisco products. Updated monthly, this online publication is organized by product category to direct you quickly to the documentation for your products. You can view the latest release of "What's New in Cisco Documentation" at this URL:

  http://www.cisco.com/univercd/cc/td/doc/abtunicd/136957.htm

- 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 Interfaces and Circuits

This chapter describes how to configure the CSS interfaces and circuits and how to bridge interfaces to Virtual LANs (VLANs). Information in this chapter applies to all CSS models, except where noted.

This chapter contains the following major sections:

Interface and Circuit Overview

The CSS provides Ethernet interfaces (ports) that enable you to connect servers, PCs, routers, and other devices to the CSS.

Using the **bridge** command, you assign the Ethernet interfaces to a specific VLAN. Each VLAN circuit requires an IP address. Assigning an IP address to each VLAN circuit allows the CSS to route Ethernet interfaces from VLAN to VLAN.

Using the **trunk** command, you can assign multiple VLANs to a CSS Ethernet interface port (Fast Ethernet port or Gigabit Ethernet port). A trunk is a point-to-point link carrying the traffic of several VLANs. The advantage of a trunk is to save ports by creating a link between two CSSs implementing VLANs. A trunk bundles virtual links over one physical link. The unique physical link between the two CSSs is able to carry traffic for the specified VLANs.

Note The **trunk** and **vlan** commands (and the associated software functionality) comply with the IEEE 802.1Q Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks.

The CSS forwards VLAN circuit traffic to the IP interface. The IP interface passes the traffic to the IP forwarding function where the CSS compares the destination of each packet to information contained in the routing table. Once the CSS resolves the packet addresses, it forwards the packet to the appropriate VLAN and destination port.

With trunking enabled, the CSS automatically inserts a tag in every frame transmitted over the trunk link to identify the originating VLAN. When the VLAN-aware CSS receives the frame, it reviews the VLAN-tagged packet to identify the transmitting VLAN. If the VLAN is recognized, the frame is routed to the proper port and VLAN destination. If the frame is from a VLAN that is not assigned to the trunk port, the packet is ignored. By default, the CSS discards untagged packets.

For an 802.1Q trunk, you can use the **default-vlan** command to:

- Accept packets that arrive untagged on the interface
- Transmit untagged packets

By using this method, the CSS can determine which VLAN transmitted an untagged frame. This capability allows VLAN-aware CSSs and VLAN-unaware CSSs to transmit and receive information on the same cable.

Figure 1-1 illustrates the interfaces, circuits, and VLANs in a CSS, and Figure 1-2 illustrates trunking between VLANs.

Figure 1-1 CSS Interfaces and Circuits

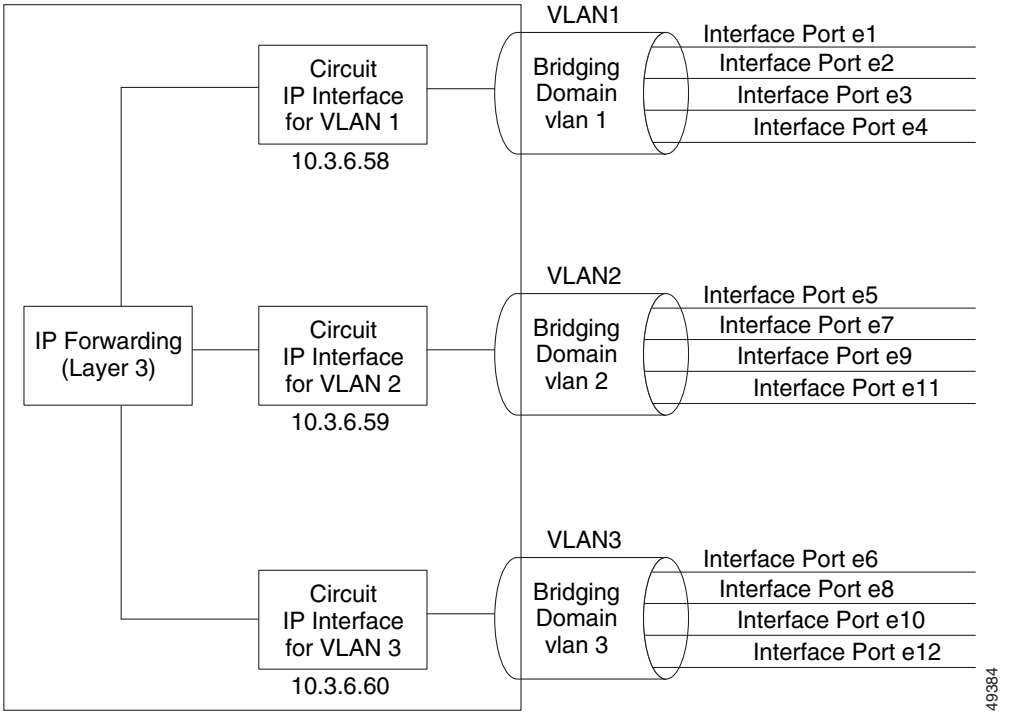
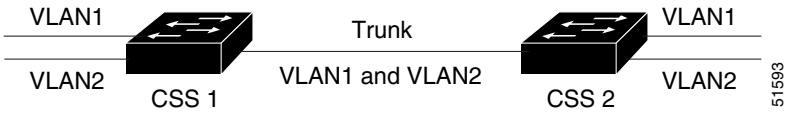


Figure 1-2 Interface Trunking Between VLANs



Interface and Circuit Configuration Quick Start

Table 1-1 provides a quick overview of the steps required to configure interfaces and circuits. 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 Interface and Circuit Configuration Quick Start

Task and Command Example
1. Log in to the CSS.

2. Enter configuration mode by typing **config**.
`# config`
`(config)#`

3. Enter the interface mode for the interface you wish to configure.
This set of interface commands applies to the CSS 11501.
`(config)# interface e1`
`(config-if[e1])#`
This set of interface commands applies to the CSS 11503 or CSS 11506.
`(config)# interface 2/1`
`(config-if[2/1])#`

4. Configure the interface duplex, speed, and flow control (default is **auto-negotiate**). `(config-if[2/1])# phy 100Mbits-FD`
5. Bridge the interface to a VLAN. All interfaces are assigned to VLAN1 by default. `(config-if[2/1])# bridge vlan 2`

6. (Optional) Enable trunking for a CSS Gigabit Ethernet or Fast Ethernet port.
`(config-if[2/1])# trunk`
`(config-if[2/1])# vlan 2`
`Create VLAN<2>, [y/n]:y`
`(config-if-vlan[2/1-2])# vlan 3`
`Create VLAN<3>, [y/n]:y`
`(config-if-vlan[2/1-3])#`

Table 1-1 Interface and Circuit Configuration Quick Start (continued)

Task and Command Example
7. (Optional) Display all circuit information for circuits that are currently active. `(config-if[2/1])# show circuit all`

8. (Optional) Display the interface configuration.
`(config-if[2/1])# show interface`
`(config-if[2/1])# exit`

9. Configure circuits as required. Assign an IP address and subnet mask to each circuit.
`(config)# circuit VLAN1`
`(config-circuit[VLAN1])# ip address 10.3.6.58/24`
`(config)# circuit VLAN3`
`(config-circuit[VLAN3])# ip address 10.3.6.60/24`
`(config-circuit-ip[VLAN3-10.3.6.60])# exit`

10. (Optional) Display the circuit configuration. `(config-circuit[VLAN1])# show circuit all`
11. (Recommended) Save your configuration changes to the startup-configuration file. If you do not save the running configuration, all configuration changes are lost upon reboot. `# copy running-config startup-config`

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

```
!********************** INTERFACE ********************
interface  2/1
  phy 100Mbits-FD
  bridge vlan 2

!*********************** CIRCUIT **********************
circuit VLAN1
    ip address 10.3.6.58 255.255.255.255

circuit VLAN3
    ip address 10.3.6.60 255.255.255.255
```

Configuring Interfaces

Interfaces are ports that enable you to connect devices to the CSS and connect the CSS to the Internet. The commands to configure interfaces on the CSS 11501 differ slightly from the commands to configure interfaces on the CSS 11503 or CSS 11506 because they require a slot/port designation. The CSS 11501 does not use the slot/port designation.

This section includes the following topics:

Configuring an Interface

To configure an Ethernet interface, use the **interface** command. Enter the interface name as follows:

- CSS 11501 - Enter the interface name in *interface port* format (for example, e1 for Ethernet interface port 1).
- CSS 11503 or CSS 11506 - Enter the interface format in *slot*/*port* format (for example, 3/1 for Ethernet port 1 on the I/O module in slot 3).

For example, to configure interface port 1 on a CSS 11501, access interface mode for the port by entering:

```
(config)# interface e1
(config-if[e1])#
```

For example, to configure interface 1 on a CSS 11503 or CSS 11506, access interface mode for the I/O module in slot 2 by entering:

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

Note in both examples that the CSS changes from configuration mode to the specific interface mode.

Entering a Description for the Interface

To identify the Ethernet interface, use the **description** command. Enter a quoted text string from 1 to 255 characters including spaces.

For example:

```
(config-if[2/1])# description "Connects to server17"
```

To view an interface description, use the **show running-config interface** command. For example:

```
(config-if[2/1])# show running-config interface 2/1

!************************ INTERFACE ************************
interface 2/1
    description "Connects to server17"
    bridge vlan 2
```

To remove an interface description, enter:

```
(config-if[2/1])# no description
```

Configuring Interface Duplex and Speed

By default, the CSS Fast Ethernet interface and Gigabit Ethernet interface are configured to auto-negotiate. The CSS automatically detects the network line speed (Fast Ethernet only) and duplex of incoming signals, and synchronizes those parameters during data transfer. Auto-negotiation enables the CSS and the other devices on the link to achieve the maximum common level of operation.

Note The CSS 1000BASE-T Gigabit Ethernet port supports 1000 Mbps full-duplex operation only and does not support auto-negotiation.

When using Fast Ethernet ports with older equipment that cannot transmit the duplex and speed with the signals, you can manually configure the speed (10 Mbps, 100 Mbps) and duplex (half or full duplex) of the CSS port to match the transmitting equipment.

When you use Gigabit Ethernet ports, if the link does not come up (perhaps due to traffic congestion), you may need to force the CSS and its link partner in to a specific mode. The CSS allows you to manually select a full duplex and flow control (pause frame) mode. Flow control allows the CSS to control traffic during congestion by notifying the other port to stop transmitting until the congestion clears. When the other device receives the pause frame, it temporarily stops transmitting data packets. When the CSS detects local congestion and becomes overwhelmed with data, the Gigabit Ethernet ports transmits a pause frame. Both the CSS Gigabit Ethernet and its link partner must be configured with the same pause method (asymmetric, symmetric, or both). By default, all Gigabit Ethernet ports are configured to full duplex mode with symmetric pause (pause frames transmitted and received by the CSS).

Note If you configure the **redundancy-phy** command on an interface of the master CSS in a box-to-box redundancy configuration and then make a 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, first enter the **no redundancy-phy** command on the interface, change the port settings, and then reenter the **redundancy-phy** command. For information about the **redundancy-phy** command, refer to the *Cisco Content Services Switch Redundancy Guide*.

Use the **phy** command to configure the duplex, speed (Fast Ethernet ports only), and flow control (Gigabit Ethernet ports only) for the interface ports, as follows:

- **phy auto-negotiate** - Resets the Fast Ethernet and Gigabit Ethernet ports to automatically negotiate port speed and duplex of incoming signals. The CSS 1000BASE-T Gigabit Ethernet port supports 1000 Mbps full-duplex operation only and does not support auto-negotiation.

Note Pause mode during auto-negotiation is not supported for the Fast Ethernet ports.

- **phy auto-negotiate** {**enable** | **disable**} - Disables the Gigabit Ethernet interface from automatically negotiating duplex of incoming signals. By default, auto-negotiation is enabled for all Gigabit Ethernet ports. The CSS 1000BASE-T port supports 1000 Mbps full-duplex operation only and does not support auto-negotiation.

  Gigabit Ethernet port auto-negotiation remains enabled when a pause mode command is specified so the Gigabit Ethernet interface ports can act upon the link partner's flow control capability. If it is necessary to disable auto-negotiation for the Gigabit Ethernet port when using a pause mode, enter the **phy auto-negotiate disable** command.
- **phy 10Mbits-FD** - Sets the Fast Ethernet port to 10 Mbps and full-duplex mode.
- **phy 10Mbits-HD** - Sets the Fast Ethernet port to 10 Mbps and half-duplex mode.
- **phy 100Mbits-FD** - Sets the Fast Ethernet port to 100 Mbps and full-duplex mode.

- **phy 100Mbits-HD** - Sets the Fast Ethernet port to 100 Mbps and half-duplex mode.
- **phy 1Gbits-FD-asym** - Sets the Gigabit Ethernet port to full-duplex mode with asymmetric pause frames transmitted toward the link partner. Asymmetric pause is useful when you need the CSS to pause its link partner but not to respond to pause frames transmitted from the link partner.
- **phy 1Gbits-FD-no pause** - Sets the Gigabit Ethernet port to full-duplex mode with no pause frames transmitted or received.
- **phy 1Gbits-FD-sym** - Sets the Gigabit Ethernet port to full-duplex mode with symmetric pause (pause frames transmitted and received by the CSS). Symmetric pause is useful for point-to-point links. By default, all Gigabit Ethernet ports are configured to full-duplex mode with symmetric pause.
- **phy 1Gbits-FD-sym-asym** - Sets the Gigabit Ethernet port to full-duplex mode with symmetric and asymmetric pause frames used with the local device.

For example, to configure Fast Ethernet interface 1 on the I/O module in slot 2 of the CSS 11503 to 100 Mbps and half-duplex mode, enter:

```
(config-if[2/1])# phy 100Mbits-HD
```

For example, to configure gigabit interface 1 on the SCM in slot 1 of the CSS 11503 to full-duplex mode with asymmetric pause, enter:

```
(config-if[1/1])# phy auto-negotiate disable
(config-if[1/1])# phy 1Gbits-FD-asym
```

Setting Interface Maximum Idle Time

As a troubleshooting tool to verify an interface's ability to receive traffic, use the **max-idle** command. If the interface does not receive traffic within the configured idle time, the CSS reinitializes the interface automatically.

Set the idle time to a value greater than the interval over which the interface is receiving traffic. For example, if the interface receives traffic every 90 seconds, set the idle time to a value greater than 90 seconds. If you set the idle time to less than 90 seconds, the CSS would continuously reinitialize the interface before the interface was able to receive traffic.

Enter an idle time from 15 to 65535 seconds. The default is 0, which disables the idle timer.

For example, to set the maximum idle time to 180 seconds for interface port 1 on a CSS 11503, the I/O module in slot 2, enter:

```
(config-if[2/1])# max-idle 180
```

To reset the idle time for an interface to its default value of 0, enter:

```
(config-if[2/1])# no max-idle
```

Bridging an Interface to a VLAN

To specify a VLAN and associate it with the specified Ethernet interface, use the **bridge vlan** command. Enter an integer from 1 to 4094 as the VLAN identifier. The default is 1. All interfaces are assigned to VLAN1 by default.

The following list defines the maximum number of VLANs supported by the specific CSS models:

- CSS 11501 and CSS 11503 - A maximum of 256 VLANs per CSS and 64 VLANs per port (FE or GE)
- CSS 11506 - A maximum of 512 VLANs per CSS and 64 VLANs per port (FE or GE)

When you specify the **bridge vlan** command, enter the word **vlan** in lowercase letters and include a space before the VLAN number (for example, **vlan 2**).

For example, to configure e1 to VLAN2 on the CSS 11501, enter:

```
(config-if[e1])# bridge vlan 2
```

The CSS Gigabit Ethernet and Fast Ethernet interface ports support trunking to multiple VLANs through the **trunk** command. In this configuration, use the **trunk** command for the Ethernet interface instead of the **bridge vlan** command (and the other associated bridge CLI commands).

To restore the default VLAN1 on the CSS 11501, enter:

```
(config-if[e7])# no bridge vlan
```

To display all interfaces and the VLANs to which they are configured, use the **show circuit** command. In the **show circuit** display, VLANs appear as VLAN (uppercase, with no space before the VLAN number). See the "Showing Circuits" section for information about the **show circuits** command.

Specifying VLAN Trunking for an Interface

To activate VLAN trunking for a CSS interface, use the **trunk** command. You specify all VLANs that include the specified port as part of the VLAN. The **trunk** command also converts the link in to a trunk link. Use the **vlan** command to specify the number of each VLAN to be associated with the Gigabit Ethernet or Fast Ethernet port. Enter an integer from 1 to 4094 as the VLAN identifier.

The following list defines the maximum number of VLANs supported by the specific CSS models:

- CSS 11501 and CSS 11503 - A maximum of 256 VLANs per CSS and 64 VLANs per port (FE or GE)
- CSS 11506 - A maximum of 512 VLANs per CSS and 64 VLANs per port (FE or GE)

The CSS software has a dependency when using the **trunk** command. For trunking to be enabled, all VLAN bridging commands for any active VLAN must first be disabled for the Gigabit Ethernet or Fast Ethernet interface by using the **no bridge vlan**, **no bridge port-priority**, **no bridge state**, and **no bridge pathcost** commands. If you do not disable VLAN bridging on an interface, the CSS software instructs you to do so.

When you specify the **trunk** command, enter the word **vlan** in lowercase letters and include a space before the VLAN number (for example, **vlan 2**). The CSS automatically prompts you to create the specified VLAN (where **y** instructs the software to create the VLAN and **n** cancels the VLAN creation).

For example, to configure Gigabit Ethernet port 1 in slot 1 for use in VLAN2, VLAN3, and VLAN9, enter:

```
(config-if[1/1])# trunk
(config-if[1/1])# vlan 2
Create VLAN<2>, [y/n]:y
(config-if-vlan[1/1-2])# vlan 3
Create VLAN<3>, [y/n]:y
(config-if-vlan[1/1-3])# vlan 9
Create VLAN<9>, [y/n]:y
(config-if-vlan[1/1-9])#
```

The **no trunk** command turns off all trunking, removes all specified **vlan** commands associated with the interface, and deletes this information from the running configuration. The interface is returned to VLAN1 by default.

To disable trunking on the specified interface and associated VLANs, enter:

```
(config-trunkif[2/3])# no trunk
```

To display all interfaces and the VLANs to which they are configured, use the **show circuit** command. In the **show circuit** output, VLANs appear as VLAN (uppercase, with no space before the VLAN number). For an interface that has trunking enabled, an "-*n*" (where *n* is the associated VLAN number) is appended to the prefix. In this example, 1/4-1 indicates slot 1, port 4, VLAN1. See the "Showing Circuits" section for information about the **show circuits** command.

Selecting a Default VLAN in a Trunk

To define a default VLAN to accept packets that arrive untagged on the interface, include the **default-vlan** command as part of the trunk/VLAN definition. The command also specifies that the packets transmitted from this VLAN will be untagged. The default VLAN must be explicitly set if you want untagged packets to be processed by the CSS. Otherwise, these packets are discarded.

The **default-vlan** command can be specified only for a single VLAN. If you attempt to use this command for another VLAN, the CSS instructs you to disable the current default VLAN using the **no default-vlan** command.

For example:

```
(config-if[1/1])# trunk
(config-if[1/1])# vlan 2
Create VLAN<2>, [y/n]:y
(config-if-vlan[1/1-2])# vlan 3
Create VLAN<3>, [y/n]:y
(config-if-vlan[1/1-3])# default-vlan
```

To remove the default VLAN selection, enter:

```
(config-if-vlan[1/1-3])# no default-vlan
```

Configuring Spanning-Tree Bridging for a VLAN or a Trunked Interface

The CSS supports configuration of Spanning-Tree Protocol (STP) bridging for an Ethernet interface in a VLAN or for a trunked Ethernet interface. Spanning-tree bridging is used to detect, and then prevent, loops in the network. You can define the bridge spanning-tree path cost, priority, and state for an Ethernet interface or for a trunked Ethernet interface. Ensure you configure the spanning-tree bridging parameters the same on all switches running STP in the network.

Note When connecting a Cisco Catalyst switch to a CSS using an 802.1Q trunk and the Spanning-Tree Protocol, the Catalyst runs a spanning-tree instance for each VLAN. When you configure an 802.1Q trunk on an Ethernet interface for the Catalyst switch, the bridge protocol data units (BPDUs) are tagged with the corresponding VLAN ID and the destination MAC address changes from the standard 01-80-C2-00-00-00 to the proprietary 01-00-0c-cc-cc-cd. This modification allows Cisco switches operating in a non-Cisco (a mix of other vendors) 802.1Q trunk environment to maintain spanning-tree states for all VLANs. Although the CSS maintains a spanning-tree instance for each VLAN as well, the CSS uses the standard 01-80-C2-00-00-00 destination MAC address for all BPDUs (tagged or untagged). When you connect a Cisco Catalyst switch to a CSS over an 802.1Q trunk, the result is that neither switch recognizes the other's BPDUs, and both assume root status. If a spanning-tree loop is detected, the Catalyst switch goes in to blocking mode on one of its looped ports.

This section includes the following topics:

- Configuring Spanning-Tree Bridge Pathcost
- Configuring Spanning-Tree Bridge Port Priority
- Configuring Spanning-Tree Bridge State

For details about globally configuring spanning-tree bridging parameters for the CSS (such as bridge aging time, forward delay time, hello time interval, and maximum age), refer to Chapter 2, Configuring Spanning-Tree Bridging for the CSS.

Configuring Spanning-Tree Bridge Pathcost

The path cost is the contribution of the interface to the vast path cost towards the spanning-tree root. Use the **bridge pathcost** command to set the spanning-tree path cost for an Ethernet interface or for a trunked Ethernet interface. Enter an integer from 1 to 65535. The default is dynamically configured based on the interface speed.

For example, to set a path cost of 9 for e7 on the CSS 11501, enter:

```
(config-if[e7])# bridge pathcost 9
```

For example, to set a path cost of 2 for the I/O module in slot 1, Ethernet port 1, in VLAN3, enter:

```
(config-if-vlan[1/1-3])# bridge pathcost 2
```

To restore the default path cost, enter:

```
(config-if-vlan[1/1-3])# no bridge pathcost
```

Configuring Spanning-Tree Bridge Port Priority

To set the spanning-tree bridge port priority for an Ethernet interface or for a trunked Ethernet interface, se the **bridge port-priority** command. If the CSS has a bridge port priority that is lower than all other switches, it will be automatically selected by the other switches as the root switch. Enter an integer from 0 to 255. The default is 128.

For example, to set a bridge port priority of 100 for e7 on the CSS 11501, enter:

```
(config-if[e7])# bridge port-priority 100
```

For example, to set a bridge port priority of 100 for the I/O module in slot 1, Ethernet port 1, in VLAN3, enter:

```
(config-if-vlan[1/1-3])# bridge port-priority 100
```

To restore the default port priority of 128, enter:

```
(config-if-vlan[1/1-3])# no bridge port-priority
```

Configuring Spanning-Tree Bridge State

By default, an Ethernet interface is set to the enabled bridge state. Use the **bridge state** command to set the spanning-tree bridge state for an Ethernet interface or for a trunked Ethernet interface.

For example, to enable the bridge state for e7 on the CSS 11501, enter:

```
(config-if[e7])# bridge state enable
```

For example, to enable the bridge state for the I/O module in slot 1, Ethernet port 1, in VLAN3, enter:

```
(config-if-vlan[1/1-3])# bridge state enable
```

To disable the bridge state, enter:

```
(config-if-vlan[1/1-3])# bridge state disable
```

Configuring Port Fast on an Interface

The Port Fast feature immediately brings a CSS Ethernet interface (port) to the Spanning Tree Protocol (STP) forwarding state from a blocking state, bypassing the listening and learning states. You can specify Port Fast for ports connected to a single workstation or server to allow those devices to immediately connect to the network, rather than waiting for the STP to converge.

Ports connected to a single workstation or server should not receive bridge protocol data units (BPDUs).

Caution The purpose of Port Fast is to minimize the time ports must wait for STP to converge. This means that the Port Fast function is effective only when used on ports connected to end stations in the network. If you enable Port Fast on a port connecting to another switch, you risk creating a spanning-tree loop. Consider using the BDPU guard feature to avoid creating a spanning-tree loop.

This section includes the following topics:

- Enabling Port Fast
- Enabling BPDU Guard
- Showing Port Fast Information

Enabling Port Fast

A port with the Port Fast feature enabled is moved directly to the spanning-tree forwarding state without waiting for the standard forward-time delay.

Caution Use Port Fast only when connecting a single end station to a CSS interface. Enabling this feature on a port connected to a switch or hub could prevent spanning tree from detecting and disabling loops in your network, which could cause broadcast storms and address-learning problems.

To enable Port Fast on a non-trunked port, use the interface mode **bridge port-fast enable** command. You cannot configure Port Fast on a trunked port. By default, Port Fast is disabled on the port.

```
(config-if[2/1])# bridge port-fast enable
```

To disable the Port Fast feature, use the interface mode **bridge port-fast disable** command.

```
(config-if[2/1])# bridge port-fast disable
```

Enabling BPDU Guard

Use the BPDU guard feature to prevent a Port Fast port on the CSS from participating in the spanning tree. When you globally enable BPDU guard on the Port Fast ports, spanning tree shuts down the ports that receive BPDUs. For information to enable Port Fast on an interface port, see the "Configuring Port Fast on an Interface" section.

In a valid configuration, the enabled Port Fast ports do not receive BPDUs. Receiving a BPDU on a Port Fast-enabled port signals an invalid configuration, such as the connection of an unauthorized device, and the BPDU guard feature puts the port in the disabled state. The BPDU guard feature provides a secure response to invalid configurations because you must manually put the port back in service.

To enable the BPDU guard on the CSS, use the global configuration **bridge bdpu-guard enabled** command:

```
(config)# bridge bpdu-guard enabled
```

To disable BPDU guard, use the global configuration **bridge bpdu-guard disabled** command:

```
(config)# bridge bpdu-guard disabled
```

Showing Port Fast Information

To display whether Port Fast is enabled or disabled on all interfaces, use the **show bridge port-fast** command. This command is available in all modes. This command also displays whether the BPDU guard feature is enabled or disabled on the CSS, and the state of the interfaces.

Table 1-2 describes the fields in the **show bridge port-fast** command output.

Table 1-2 Field Description for the show bridge port-fast Command

Field	Description
BPDU guard is *state* on this switch.	The state of the BPDU guard feature on the CSS: Enabled or Disabled.
Name	The number of the module slot and interface.
IfIndex	The interface index number.
Type	The type of interface. • **fe** indicates a Fast Ethernet interface. • **ge** indicates a Gigabit Ethernet interface.
Oper	The operational state of the interface: Up or Down.
Admin	The administration state: Enable or Down.
PortFast State	Indicates whether Port Fast is enabled or disabled on the interface.

Showing Interface Configurations

This CSS includes a series of **show** interface mode commands that enable you to view interface configuration information about the CSS. This information includes VLAN bridging, VLAN trunk status, list of valid Ethernet interfaces, interface duplex and speed values, interface statistics, and errors on an Ethernet interface.

This section includes the following topics:

Showing Bridge Configurations

The CSS enables you to show bridging information for a specific VLAN in the CSS. Use the **show bridge** command to display this bridging information.

The syntax for this command is:

show bridge [**forwarding**|**status**] {*vlan_number*}

The options and variables are as follows:

- **forwarding** - Displays the bridge forwarding table including the VLAN number, the MAC addresses, and port numbers.
- **status** - Displays the bridge spanning-tree status including the Spanning Tree Protocol (STP) state; designated root; bridge ID; root maximum age; hello time and forward delay; and port information including state, VLAN, root and port cost, and designated root and port number.
- *vlan_number* - Displays the forwarding table or spanning tree status for the specified VLAN number. To see a list of VLAN numbers, enter **show bridge** [**forwarding**|**status**] **?**

To display bridge forwarding or bridge status for a specific VLAN in the CSS, enter the **show bridge forwarding** or the **show bridge status** command with the VLAN number. Entering the **show bridge** command with a VLAN number returns a list of available VLANs.

Table 1-3 describes the fields in the **show bridge forwarding** command output.

Table 1-3 Field Descriptions for the show bridge forwarding Command

Field	Description
VLAN	The bridge interface virtual LAN number
MAC Address	The MAC address for the entries
Port Number	The port number for the bridge forwarding table

Table 1-4 describes the fields in the **show bridge status** command output.

Table 1-4 Field Descriptions for the show bridge status Command

Field	Description
STP State	The state of the Spanning-Tree Protocol: Enabled or Disabled.
Root Max Age	The timeout period, in seconds, during which the host times out root information.
Root Hello Time	The interval, in seconds, that the root bridge broadcasts its hello message to other CSSs.
Root Fwd Delay	The delay time, in seconds, that the root bridge uses for forward delay.
Designated Root	The bridge ID for the designated root.
Bridge ID	The bridge ID of this bridge.
Port	The port ID.

Table 1-4 Field Descriptions for the show bridge status Command (continued)

Field	Description
State	The state of the port. The possible states are as follows: • Block - The blocking state. A port enters the blocking state after CSS initialization. The port does not participate in frame forwarding. • Listen - The listening state. This state is the first transitional state a port enters after the blocking state. The port enters this state when STP determines that the port should participate in frame forwarding. • Learn - The learning state. The port enters the learning state from the listening state. The port in the learning state prepares to participate in frame forwarding. • Forward - The forwarding state. The port enters the forwarding state from the learning state. A port in the forwarding state forwards frames. • Disabled - The disabled state. A port in the disabled state does not participate in frame forwarding or STP. A port in the disabled state is nonoperational.
Designated Bridge	The bridge ID for the designated bridge.
Designated Root	The bridge ID for the designated root.
Root Cost	The cost of the root.
Port Cost	The cost of the port.
Desg Port	Designated port.

Showing Trunking Configurations

The CSS enables you to show VLAN trunk status information for Gigabit Ethernet and Fast Ethernet ports. To display this information, use the **show trunk** command.

Table 1-5 describes the fields in the **show trunk** command output.

Table 1-5 Field Descriptions for the show trunk Command

Field	Description
Port	The CSS port
VLAN	The VLAN on the port
Default VLAN	The configured default VLAN on the port (if there is no configured default VLAN, “None” appears in this field)

Showing Interface Information

To display a list of valid interfaces for the CSS, use the **show interface** command. For example:

```
(config)# show interface
```

To display information for a specific interface, enter the **show interface** command and the interface name. Enter the interface name as follows:

- CSS 11501 - Enter the interface name in *interface port* format (for example, e1 for Ethernet interface port 1).
- CSS 11503 or CSS 11506 - Enter the interface format in *slot*/*port* format (for example, 3/1 for Ethernet port 1 on the I/O module in slot 3).

For example, to show interface information for port 1 on a CSS 11503, the I/O module in slot 2, enter:

```
(config)# show interface 2/1
```

Table 1-6 describes the fields in the **show interface** command output.

Table 1-6 Field Descriptions for the show interface Command

Field	Description
Name	The name of the interface.
ifIndex	The Index for the interface.
Type	The type of interface. The possible types include: • fe - Fast Ethernet interface • ge - Gigabit Ethernet interface • console - Console interface
Oper	Operational state: Up or Down.
Admin	Administrative state: Up or Down.
Last Change	The date of the last state change.

Showing Interface Duplex and Speed

Use the **show phy** command to show duplex and speed values for all interfaces. For example:

```
(config)# show phy
```

To show duplex and speed value for a specific interface, specify the **show phy** command and the interface name. Enter the interface name as follows:

- CSS 11501 - Enter the interface name in *interface port* format (for example, e1 for Ethernet interface port 1).
- CSS 11503 or CSS 11506 - Enter the interface format in *slot*/*port* format (for example, 3/1 for Ethernet port 1 on the I/O module in slot 3).

For example, to show the interface and duplex speed for interface port 1 on a CSS 11506, the I/O module in slot 2, enter:

```
(config)# show phy 2/1
```

Table 1-7 describes the fields in the **show phy** command output.

Table 1-7 Field Descriptions for the show phy Command

Field	Description
Name	The name of the physical interface.
Configured Speed	The configured speed for the Ethernet interface (port) in the CSS. Auto indicates the speed is automatically negotiated.
Configured Duplex	The configured duplex for the Ethernet interface (port) in the CSS. Auto indicates the duplex is automatically negotiated.
Actual Speed	The actual speed for the Ethernet interface (port) in the CSS.
Actual Duplex	The configure duplex for the Ethernet interface (port) in the CSS.
Link	The link status: Up or Down.
Rev	Revision number of the chip.
Partner Auto	Indicates whether auto-negotiation is available on the link partner.

Showing Interface Statistics

Use the **show mibii** command to display the extended 64-bit MIB-II statistics for a specific interface, or for all interfaces in the CSS. The CSS Enterprise ap64Stats MIB defines these statistics. The Gigabit Ethernet module port statistics are an aggregation of all ports on the module.

To display the RFC 1213 32-bit statistics, include the **-32** suffix.

To display extended MIB-II statistics for a specific interface in the CSS, enter the **show mibii** command with the interface name. To see a list of interfaces in the CSS, enter **show mibii ?**.

Note Refer to the *Cisco Content Services Switch Administration Guide* for information on CSS MIBs.

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

Table 1-8 Field Descriptions for the show mibii Command

Field	Description
MAC	The interface address at the protocol layer immediately below the network layer in the protocol stack. For interfaces that do not have such an address (for example, a serial line), this object contains an octet string of zero length.
Administrative	The desired state of the interface (Enabled, Disabled, or Testing). The testing state indicates no operational packets can be passed.
MTU	The size of the largest datagram that can be sent or received on the interface, specified in octets. For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the interface.
In Octets	The total number of octets received on the interface, including framing characters.
In Unicast	The number of subnetwork-unicast packets delivered to a higher-layer protocol.
In Multicast	The number of non-unicast (for example, subnetwork-broadcast or subnetwork-multicast) packets delivered to a higher-layer protocol.
In Errors	The number of inbound packets that contained errors preventing them from being delivered to a higher-layer protocol.
In Discards	The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being delivered to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.
In Unknown	The number of packets received over the interface that were discarded because of an unknown or unsupported protocol.

Table 1-8 Field Descriptions for the show mibii Command (continued)

Field	Description
Last Change	The value of sysUpTime at the time the interface entered its current operational state. If the state has not changed since the time the CSS came up, the sysUptime is when the port was initialized.
Operational	The current operational state of the interface (Up, Down, or Testing). The Testing state indicates no operational packets can be passed.
Speed	An estimate of the interface's current bandwidth, in bits per second. For interfaces that do not vary in bandwidth or for those where no accurate estimation can be made, this object contains the nominal bandwidth.
Queue Len	The length of the output packet queue (in packets).
Out Octets	The total number of octets transmitted out of the interface, including framing characters.
Out Unicast	The total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including those packets that were discarded or not sent.
Out Multicast	The total number of packets that higher-level protocols requested be transmitted to a non-unicast (for example, a subnetwork-broadcast or subnetwork-multicast) address, including those packets that were discarded or not sent.
Out Errors	The number of outbound packets that could not be transmitted because of errors.
Out Discards	The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.

To clear interface statistics, use the **clear statistics** command in SuperUser mode. For example:

```
# clear statistics
```

Showing Ethernet Interface Errors

To list the errors on an Ethernet interface, use the **show ether-errors** command and options. When required, enter the interface name as a case-sensitive unquoted text string. To see a list of interfaces, enter **show ether-errors ?**.

The command provides the following options:

- **show ether-errors** - Displays the extended 64-bit statistics for errors on all Ethernet interfaces in the CSS. The Enterprise ap64Stats MIB defines these statistics.
- **show ether-errors** *interface name* - Displays the extended 64-bit statistics for errors on a specific Ethernet interface in the CSS. The Enterprise ap64Stats MIB defines these statistics. Enter the interface name as a case-sensitive unquoted text string.
- **show ether-errors zero** - Displays the Ethernet errors for all Ethernet interfaces in the CSS and reset the statistics to zero upon retrieval.
- **show ether-errors zero** *interface name* - Displays the Ethernet errors for the specified Ethernet interface in the CSS and resets the statistics to zero upon retrieval. Enter the interface name as a case-sensitive unquoted text string.
- **show ether-errors-32** - Displays the RFC 1398 32-bit statistics, including the **-32** suffix.
- **show ether-errors-32** *interface name* - Displays the RFC 1398 32-bit statistics, including the **-32** suffix. Enter the interface name as a case-sensitive unquoted text string.

Table 1-9 describes the fields in the **show ether-errors** command output.

Table 1-9 Field Descriptions for the show ether-errors Command

Field	Description
Alignment	The number of frames with alignment errors (frames that do not end with a whole number of octets and have a bad cyclic redundancy check) received on the interface.
FCS	The number of frames received on the interface that are an integral number of octets in length but do not pass the frame check sequence (FCS) check.

Table 1-9 Field Descriptions for the show ether-errors Command (continued)

Field	Description
Single Collision	The number of successfully transmitted frames on the interface for transmissions that were inhibited by exactly one collision.
Multiple Collisions	The number of successfully transmitted frames on the interface for transmissions that were inhibited by more than one collision.
SQE Test	The number of times that the SQE TEST ERROR message is generated.
Deferred Tx	The number of frames for which the first transmission attempt on the interface is delayed because the medium is busy. The count represented by an instance of this object does not include frames involved in collisions.
Internal Rx Errors	The number of frames for which reception on the interface failed due to an internal MAC sublayer receive error.
Frame too Long	The number of frames received on the interface that exceeded the maximum permitted frame size.
Carrier Sense Errors	The number of times that the carrier sense condition was lost or never asserted when attempting to transmit a frame on the interface.
Internal Tx Errors	The number of frames for which transmission on the interface failed due to an internal MAC sublayer transmit error.
Excessive Collisions	The number of frames for which transmission on the interface failed due to excessive collisions.
Late Collisions	The number of times that a collision is detected on the interface later than 512 bit-times in to the transmission of a packet.

Shutting Down an Interface

To shut down an interface, use the **admin-shutdown** or **shut** command.

Caution Shutting down an interface terminates all connections to the interface.

For example:

- To shut down interface e3 on the CSS 11501 with the **admin-shutdown** command, enter:

```
(config-if[e3])# admin-shutdown
```

- To shut down interface e3 on the CSS 11501 with the **shut** command, enter:

```
(config-if[e3])# shut
```

  When you use the **shut** command, the CSS changes the **shut** command to the **admin-shutdown** command in the running configuration.

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.

Shutting Down All Interfaces

To shut down all interfaces simultaneously, use the **admin-shutdown** command. This command is only available in the SuperUser mode. The **admin-shutdown** command provides a quick way to shut down all physical devices in the CSS.

Caution Shutting down an interface terminates all connections to the interface.

To shut down all interfaces, enter:

```
# admin-shutdown
```

Restarting an Interface

To restart an interface, use the **no admin-shutdown** or **no shut** command. For example:

- To restart interface e3 on the CSS 11501 with the **no admin-shutdown** command, enter:

  ```
  (config-if[e3])# no admin-shutdown
  ```

- To restart interface e3 on the CSS 11501 with the **no shut** command, enter:

  ```
  (config-if[e3])# no admin-shutdown
  ```

Note The CSS automatically sends a gratuitous ARP for the IP interface address when you restart the interface. The gratuitous ARP informs all network nodes about ARP mapping. The CSS transmits one ARP request packet and one ARP reply packet for every gratuitous ARP invocation.

Restarting All Interfaces

To restart all interfaces, enter:

```
# no admin-shutdown
```

Note The CSS automatically sends a gratuitous ARP for every configured IP interface address when you restart all interfaces. The gratuitous ARP informs all network nodes about ARP mapping. The CSS transmits one ARP request packet and one ARP reply packet for every gratuitous ARP invocation.

Configuring Circuits

A circuit on the CSS is a logical entity that maps IP interfaces to a logical port or group of logical ports, for example, a VLAN. Each VLAN circuit requires an IP address. Assigning an IP address to each VLAN circuit allows the CSS to route Ethernet interfaces from VLAN to VLAN. Router Discovery Protocol (RDP) settings can also be configured for each circuit VLAN to advertise the CSS to hosts.

This section includes the following topics:

Entering Circuit Configuration Mode

To enter the circuit configuration mode to configure a VLAN, use the **circuit** command. Enter the specific VLAN in uppercase letters. Do not include a space between VLAN and the VLAN number. For example:

```
(config)# circuit VLAN7
(config-circuit[VLAN7])#
```

Configuring a Circuit IP Interface

This section includes the following topics:

Configuring a Circuit IP Address

To assign an IP address to a circuit, use the **ip address** command. Enter the IP address and a subnet mask in CIDR bit-count notation or a mask in dotted-decimal notation. The subnet mask range is 8 to 31.

For example, to configure an IP address and subnet mask for VLAN7, enter:

```
(config-circuit[VLAN7])# ip address 172.16.6.58/8
```

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

```
(config-circuit-ip[VLAN7-172.16.6.58])#
```

Note The CSS automatically sends a gratuitous ARP for the IP interface address when you assign an IP address to a circuit. The gratuitous ARP informs all network nodes about ARP mapping. The CSS transmits one ARP request packet and one ARP reply packet for every gratuitous ARP invocation.

To remove a local IP address from a circuit, enter the following command from circuit mode:

```
(config-circuit[VLAN7])# no ip address
```

Configuring a Circuit-IP Broadcast Address

To change the broadcast address associated with a circuit, use the **broadcast** command. If you leave the broadcast address at zero, the all-ones host is used for numbered interfaces.

The default broadcast address is an all-ones host address (for example, IP address 172.16.6.58/24 has a broadcast address of 172.16.6.58/255). This command is available in IP configuration mode.

For example, to change the broadcast address on circuit VLAN7, enter:

```
(config-circuit-ip[VLAN7-172.16.6.58])# broadcast 0.0.0.0
```

To reset the broadcast IP address to the default all-ones host address, enter:

```
(config-circuit[VLAN7-172.16.6.58])# no broadcast
```

Configuring Circuit-IP Redirects

By default, the transmission of Internet Control Message Protocol (ICMP) redirect messages is enabled. To disable the transmission of ICMP redirect messages, enter:

```
(config-circuit-ip[VLAN7-172.16.6.58])# no redirects
```

To reenable the transmission of ICMP redirect messages, use the **redirects** command. For example:

```
(config-circuit-ip[VLAN7-172.16.6.58])# redirects
```

Configuring Circuit-IP Unreachables

By default, the transmission of ICMP Destination Unreachable is enabled. To disable the transmission of ICMP Destination Unreachable messages, enter:

```
(config-circuit-ip[VLAN7-172.16.6.58])# no unreachables
```

Use the **unreachables** command to enable the transmission of ICMP Destination Unreachable messages. The default state is enabled.

For example:

```
(config-circuit-ip[VLAN7-172.16.6.58])# unreachables
```

Configuring Router-Discovery Preference for a Circuit IP Interface

To enable router discovery and configure the router discovery preference value for a circuit IP interface, use the **router-discovery** command. When enabled, router discovery transmits packets with the "all-hosts" multicast address of 244.0.0.1.

Note To enable an interface to transmit packets with the limited broadcast multicast address of 255.255.255.255, use the **router-discovery limited-broadcast** command in circuit mode (see the "Configuring Router-Discovery Limited-Broadcast" section). Router discovery is disabled by default.

Use the **router-discovery preference** command to specify the preference level for the advertised CSS circuit IP address, relative to other devices on the same network. The value is an integer from 0 (default) to 65535. If you use the default value, you do not need to use this command.

For example, to specify a router discovery preference value of 100, enter:

```
(config-circuit-ip[VLAN7-192.168.1.58])# router-discovery
(config-circuit-ip[VLAN7-192.168.1.58])# router-discovery preference
100
```

To disable router discovery, enter:

```
(config-circuit-ip[VLAN7-192.168.1.58])# no router-discovery
```

To restore the router discovery preference value to the default of 0, enter:

```
(config-circuit-ip[VLAN7-192.168.1.58])# no router-discovery
preference
```

Enabling and Disabling a Circuit IP

By default, the IP interface on a circuit is enabled. To disable the IP interfaces on a circuit, enter:

```
(config-circuit-ip[VLAN7-172.16.6.58])# no enable
```

To reenable the IP interface on a circuit, use the **enable** command. For example:

```
(config-circuit-ip[VLAN7-172.16.6.58])# enable
```

Configuring Router-Discovery Protocol Settings for a Circuit

The CSS allows you to enable Router Discovery Protocol (RDP) settings and define a router discovery preference for each circuit VLAN. RDP announces the existence of the CSS to hosts by periodically multicasting or broadcasting a router advertisement to each interface.

Use the **circuit** command to enter the circuit configuration mode before configuring RDP for a circuit VLAN.

This section includes the following topics:

Configuring the Router-Discovery Lifetime

By default, the maximum age that hosts remember router advertisements is three times the **max-advertisement-interval**. Use the **router-discovery lifetime** command to configure the maximum age, in seconds. Enter an integer between 0 and 9000 seconds.

For example:

```
(config-circuit[VLAN7])# router-discovery lifetime 600
```

To reset the time to the default of three times the **max-advertisement-interval**, enter:

```
(config-circuit[VLAN7)# no router-discovery lifetime
```

Configuring Router-Discovery Limited-Broadcast

By default, the CSS transmits router discovery packets using the limited broadcast address is 224.0.0.1 (the “all-hosts” multicast address). Use the **router-discovery limited-broadcast** command to transmit router discovery packets using the limited broadcast address 255.255.255.255.

For example:

```
(config-circuit[VLAN7])# router-discovery limited-broadcast
```

To revert to the default of 224.0.0.1, enter:

```
(config-circuit[VLAN7)# no router-discovery limited-broadcast
```

Configuring the Router-Discovery Max-Advertisement-Interval

By default, the maximum interval timer used for router discovery advertisement from the circuit VLAN is 600 (10 minutes). Use the **router-discovery max-advertisement-interval** command to configure the maximum interval timer used for router discovery advertisement from the circuit VLAN. This command defines the maximum interval, in seconds, between sending advertisements. Enter an integer from 4 to 1800.

For example:

```
(config-circuit[VLAN7])# router-discovery max-advertisement-interval
300
```

To restore the router discovery maximum advertisement interval to the default of 600, enter:

```
(config-circuit[VLAN7])# no router-discovery
max-advertisement-interval
```

Configuring the Router-Discovery Min-Advertisement-Interval

By default, the minimum router advertisement interval is 0.75 times the maximum advertisement value. To configure the minimum interval timer used for router discovery advertisement from the circuit VLAN, use the **router-discovery min-advertisement-interval** command. This command defines the minimum interval, in seconds, between sending advertisements. Enter an integer from 0 to 1800.

The default is 0.75 times the max-advertisement-interval. If this value is greater than 0, it must be less than the value specified using the **router-discovery max-advertisement-interval** command.

For example:

```
(config-circuit[VLAN7])# router-discovery min-advertisement-interval
100
```

To reset the minimum router advertisement interval to the default of 0.75 times the maximum advertisement value, enter:

```
(config-circuit[VLAN7])# no router-discovery
min-advertisement-interval
```

Showing Circuits

Use the **show circuits** command to show circuit information. This command provides the following options:

- **show circuits** - Displays all circuit information for circuits that are currently up
- **show circuits all** - Displays all circuit information regardless of circuit state
- **show circuit name** *circuit name* - Displays circuit information for a specific circuit regardless of state

To list all circuits and their interfaces in the Up state, enter:

```
# show circuits
```

To list all circuits and their interfaces regardless of their state, enter:

```
# show circuits all
```

To list an individual circuit, enter:

```
# show circuits name VLAN5
```

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

Table 1-10 Field Descriptions for the show circuits Command

Field	Description
Circuit Name	The circuit name. The VLAN name appear in uppercase, with no space before the VLAN number.
Circuit State	The state of the circuit. The possible states are as follows: • active-ipEnabled • down-ipEnabled • active-ipDisabled • down-ipDisabled
IP Address	IP interface address.
Interface(s)	The interface associated with the circuit.
Operational Status	The operational status of the interface (Up or Down).

Showing IP Interfaces

Use the **show ip interfaces** command to display configured IP interfaces on the CSS. The display includes the circuit state, IP address, broadcast address, Internet Control Message Protocol (ICMP) settings, and Router Discovery Program (RDP) settings. For example:

```
# show ip interfaces
```

Table 1-11 describes the fields in the **show ip interfaces** command output.

Table 1-11 Field Descriptions for the show ip interfaces Command

Field	Description
Circuit Name	The name of the circuit associated with the IP interface.
State	The state of the IP interface. The possible states are as follows: • **Active (1)** - The interface is up • **Disabled** - The interface is disabled • **NoCircuit** - The interface is waiting for an underlying circuit
IP Address	The IP address assigned to the circuit.
Network Mask	The network mask of the circuit.
Broadcast Address	The broadcast IP address associated with the IP interface. If left at zero, the all-ones host is used for numbered interfaces. 255.255.255.255 is always used for unnumbered interfaces.
Redundancy	Indicates whether the redundancy protocol is running on the interface. The default state is Disabled.
ICMP Redirect	Indicates whether the transmission of Internet Control Message Protocol (ICMP) redirect messages is Enabled or Disabled. The default state is Enabled.
ICMP Unreachable	Indicates whether the transmission of ICMP Destination Unreachable messages is enabled or disabled. The default state is Enabled.
RIP	Indicates whether RIP is Enabled or Disabled.

Configuring RIP for an IP Interface

You can configure Routing Information Protocol (RIP) attributes on each IP interface. To configure RIP parameters and run RIP on an IP interface, use the following routing commands within the specific circuit IP mode. The default mode is to send RIP version 2 (v2) and receive either RIP or RIP2.

The timers used by RIP in the CSS include the following default values. These RIP timer values are not user-configurable in the CSS.

- Transmit (Tx) time that is a random value between 15 and 45 seconds to avoid router synchronization problems
- Route expiration time of 180 seconds (if the CSS loses the link to the next hop router, the route is immediately removed)
- Hold-down time (the amount of time the CSS transmits with an infinite metric) of 120 seconds

This section includes the following topics:

Enabling RIP on an IP Interface

To start running RIP on an IP interface, use the **rip** command. For example:

```
(config-circuit-ip[VLAN7-192.168.1.58)# rip
```

To stop running the RIP on the interface, enter:

```
(config-circuit-ip[VLAN7-192.168.1.58])# no rip
```

Configuring a RIP Default Route

By default, the CSS advertises a default route on an IP interface with a metric of 1. To advertise a default route on an IP interface with a specific metric, use the **rip default-route** command. You can also specify an optional metric in the command line. The CSS uses this metric when advertising a route. Enter a number from 1 to 15.

For example:

```
(config-circuit-ip[VLAN7-192.168.1.58])# rip default-route 9
```

Configuring a RIP Receive Version

By default, the interface receives both RIP version 1 and RIP version 2. To specify the RIP version that the interface receive, use the **rip receive** command. The options for this command are as follows:

- **rip receive both** - Receives both RIP version 1 and RIP version 2 (default)
- **rip receive none** - Receives no RIP packets
- **rip receive v1** - Receives RIP version 1 packets only
- **rip receive v2** - Receives RIP version 2 packets only

For example:

```
(config-circuit-ip[VLAN7-192.168.1.58])# rip receive both
```

Configuring RIP Send Version

By default, the interface sends RIP version 2 packets only. To specify the RIP version that the interface transmits, use the **rip send** command. The options for this command are as follows:

- **rip send none** - Sends no RIP packets
- **rip send v1** - Sends RIP version 1 packets only
- **rip send v2** - Sends RIP version 2 packets only (default)

For example:

```
(config-circuit-ip[VLAN7-192.168.1.58])# rip send v1
```

Configuring RIP Packet Logging

By default, CSS of logging received or transmitted RIP packets on the interface is disabled. Use the **rip log** command to enable the CSS to log received or transmitted RIP packets on the interface.

The options for this command are as follows:

- **rip log rx** - CSS logs RIP packets received on the interface
- **rip log tx** - CSS logs RIP packets transmitted on the interface

For example:

```
(config-circuit-ip[VLAN7-192.168.1.58])# rip log rx
```

Showing RIP Configurations for IP Addresses

Use the **show rip** command to show a RIP configuration for one IP address or all IP addresses configured in the CSS. The options for this command are as follows:

- **show rip** - Displays RIP configurations for all interfaces (including the logging of RIP packets)
- **show rip** *ip_address* - Displays a single RIP interface entry
- **show rip globals** - Displays RIP global statistics
- **show rip statistics** - Displays RIP interface statistics for all interfaces
- **show rip statistics** *ip_address* - Displays RIP interface statistics for a specific interface

Table 1-12 describes the fields in the **show rip** command output.

Table 1-12 Field Descriptions for the show rip Command

Field	Description
IP Address	The advertised RIP interface address.
State	The operational state of the RIP interface.
RIP Send	The RIP version that the interface sends. The possible values are as follows: • **none** - Do not send RIP packets • **RIPv1** - Send RIP version 1 packets only • **RIPv2** - Send RIP version 2 packets only (default)
RIP Recv	The RIP version that the interface receives. The possible values are as follows: • **both** - Receiving both version 1 and version 2 (default) • **none** - Receiving no RIP packets • **Ripv1** - Receiving RIP version 1 packets only • **Ripv2** - Receiving RIP version 2 packets only
Default Metric	The default metric used when advertising the RIP interface.
Tx Log	The setting for the logging of RIP packet transmissions (Enabled or Disabled). The default setting is disabled.
Rx Log	The setting for the logging of RIP packet received (Enabled or Disabled). The default setting is disabled.

To display global RIP statistics, enter:

```
# show rip globals
```

Table 1-13 describes the fields in the **show rip globals** command output.

Table 1-13 Field Descriptions for the show rip globals Command

Field	Description
RIP Route Changes	The global number of route changes made to the IP route database by RIP
RIP Query Responses	The global number of query responses sent to RIP query from other systems

To display the RIP interface statistics for all RIP interface entries, enter:

```
# show rip statistics
```

Table 1-14 describes the fields in the **show rip statistics** command output.

Table 1-14 Field Descriptions for the show rip statistics Command

Field	Description
System Route Changes	The global number of route changes made to the IP route database by RIP
System Global Query Responses	The global number of query responses sent to RIP query from other systems
IP Address	The RIP interface IP address
Triggered Updates Sent	The number of triggered RIP updates sent by the interface
Bad Packets Received	The number of bad RIP response packets received by the interface
Bad Routes Received	The number of bad routes in valid RIP packets received by the interface

Configuring the Switched Port Analyzer Feature

Configure the switched port analyzer (SPAN) feature on your CSS to mirror (copy) traffic passing through one CSS port (Fast Ethernet or Gigabit Ethernet) to another designated port of the same type and on the same CSS module for analysis. You can use SPAN for network troubleshooting or tuning using a network analyzer. SPAN is sometimes referred to as *port mirroring* or *port monitoring*.

A SPAN session is the association of a destination port with a source port on the same CSS module. The port that is monitored is called the source SPAN (SSPAN) port. An SSPAN port consists of two components:

- Ingress path - Network traffic entering the CSS. The CSS copies to the monitoring port packets that the SSPAN port receives (SSPAN Rx) from the network.
- Egress path - Network traffic leaving the CSS. The CSS copies to the monitoring port packets that the SSPAN port transmits (SSPAN Tx) to the network.

SPAN can monitor the ingress path, the egress path, or both. You can configure only one SSPAN port in a CSS chassis.

The port that monitors the SSPAN port is called the destination SPAN (DSPAN) port. You can configure only one DSPAN port in a CSS chassis and it must have the following characteristics:

- Same speed as the SSPAN port
- Same media type as the SSPAN port
- Local (physically resides on the same CSS module)

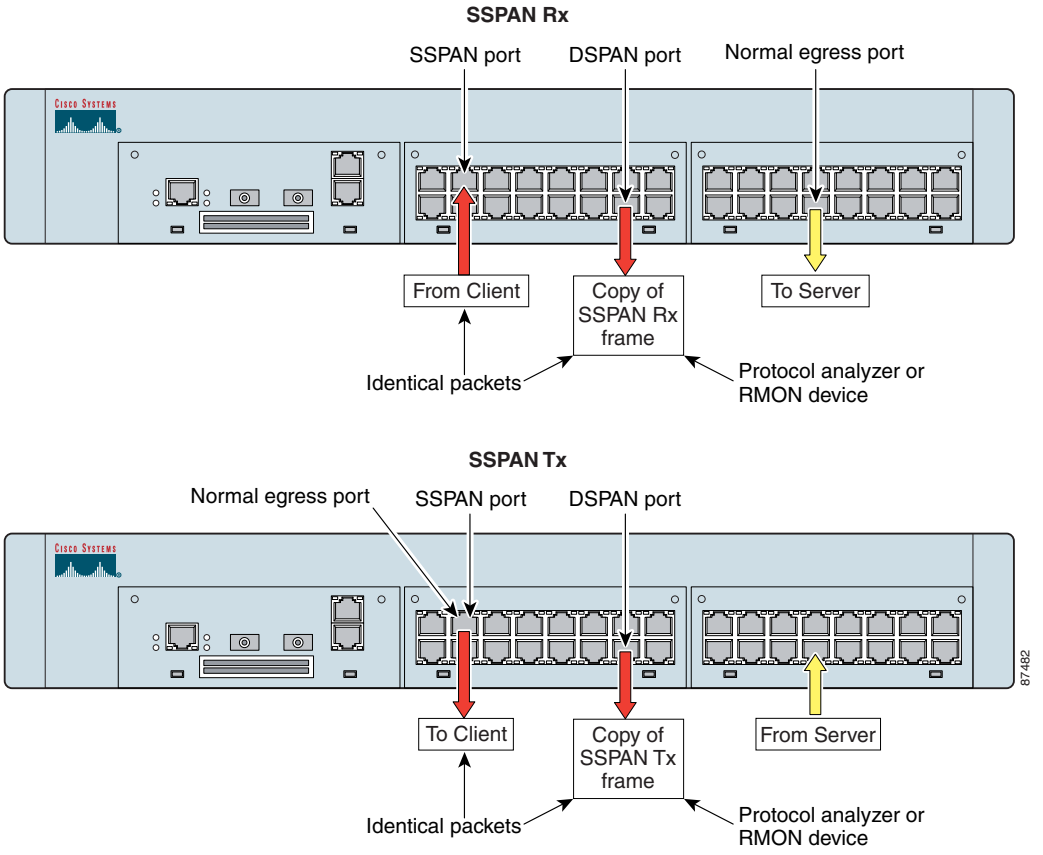
Once you configure a port as a DSPAN port, the CSS removes it from all VLANs and ignores ingress traffic on that port. In addition, the DSPAN port does not participate in STP or routing protocols such as RIP and OSPF.

Traffic copied to the DSPAN port is typically forwarded to a network analyzer, protocol analyzer, or an RMON probe. SPAN allows you to monitor CSS ports without:

- Disconnecting cables
- Requiring multiple analyzers or probes
- Needing hubs or switches

Figure 1-3 shows an example of SPAN connectivity with a protocol analyzer connected to port 2/13 on a CSS. In this example, the CSS copies all packets received or transmitted on Fast Ethernet (FE) port 2/4 (SSPAN port) to FE port 2/13 (DSPAN port). The analyzer connected to DSPAN port 2/13 receives all network traffic that the SSPAN port receives or transmits.

Figure 1-3 Example of SPAN Connectivity



This section describes how to configure SPAN on a CSS. It includes the following topics:

Configuring SPAN on a CSS

To configure SPAN on a CSS, use the **setspan** command. This command instructs the CSS to monitor all incoming and/or outgoing traffic on a specified SSPAN port by copying the packets to a specified DSPAN port on the same module in the CSS. This feature is disabled by default.

The syntax of this global configuration mode command is:

setspan src_port *number* **dest_port** *number*
copyBoth|copyTxOnly|copyRxOnly

The options and variables for this command are as follows:

- **src_port** *number* - Source port keyword and number of the SSPAN port (in slot/port format) that you want to monitor. The CSS copies all packets that are received or transmitted on this port to the DSPAN port.
- **dest_port** *number* - Destination port keyword and number of the DSPAN port (in slot/port format) where you want to connect the network analyzer, protocol analyzer, or RMON probe. The CSS copies the packets that flow through the SSPAN port to the DSPAN port that you specify. The DSPAN port must reside on the same module as the SSPAN port.

  Note Once you configure a port as a DSPAN port, the CSS removes it from all VLANs and ignores ingress traffic on that port. In addition, the DSPAN port does not participate in spanning tree protocol (STP) or routing protocols such as RIP and OSPF.

- **copyBoth** - CSS copies to the DSPAN port packets that the SSPAN port transmits to the network (egress traffic) and packets that the SSPAN port receives from the network (ingress traffic).

Note If the combined traffic bandwidth of the ingress and egress traffic of the SSPAN port exceeds the bandwidth of the DSPAN port, the DSPAN port may become oversubscribed.

- **copyTxOnly** - CSS copies to the DSPAN port only those packets that the SSPAN port transmits to the network (egress traffic).
- **copyRxOnly** - CSS copies to the DSPAN port only those packets that the SSPAN port receives from the network (ingress traffic).

For example, to copy all received and transmitted packets on SSPAN port 3 of the I/O module in slot 3 to DSPAN port 12 on the same module, enter:

```
(config)# setspan src_port 3/3 dest_port 3/12 copyBoth
```

To return the SPAN feature to its default state of disabled, use the **no setspan** command. For example, to disable SPAN on the source and destination ports on CSS module 3 in the example above, enter:

```
(config)# no setspan src_port 3/3 dest_port 3/12
```

Verifying the SPAN Configuration on a CSS

To verify the SPAN configuration on a CSS, use the **show setspan** command. Table 1-15 describes the fields in the **show setspan** command output.

Table 1-15 Field Descriptions for the show setspan Command

Field	Description
SPAN Configuration	
Source	Number of the SSPAN port whose traffic you want to monitor.
Destination	Number of the DSPAN port to which the CSS copies the packets flowing through the SSPAN port. Connect the network analyzer or RMON probe to this port.

Table 1-15 Field Descriptions for the show setspan Command (continued)

Field	Description
Direction	Direction of the traffic that you want to monitor at the source port. The direction can be one of the following: • **copyBoth** - The CSS copies packets that are transmitted and received by the SSPAN port to the DSPAN port. • **copyTxOnly** - The CSS copies only packets transmitted (egress traffic) by the SSPAN port to the DSPAN port. • **copyRxOnly** - The CSS copies only packets received (ingress traffic) by the SSPAN port to the DSPAN port.

CHAPTER 2

Configuring Spanning-Tree Bridging for the CSS

The CSS supports configuration of Spanning-Tree Protocol (STP) bridging. Spanning-tree bridging detects, and then prevents, loops in the network. Use the **bridge** command to configure global spanning-tree bridging options for the CSS, such as bridge aging time, forward delay time, hello time interval, and maximum age. Make sure you configure the spanning-tree bridging parameters the same on all switches running STP in the network.

Note When connecting a Cisco Catalyst switch to a CSS using an 802.1Q trunk and the STP, the Catalyst runs a spanning-tree instance for each VLAN. When you configure an 802.1Q trunk on an Ethernet interface for the Catalyst switch, the bridge protocol data units (BPDUs) are tagged with the corresponding VLAN ID and the destination MAC address changes from the standard 01-80-C2-00-00-00 to the proprietary 01-00-0c-cc-cc-cd. This modification allows Cisco switches operating in a non-Cisco (a mix of other vendors) 802.1Q trunk environment to maintain spanning-tree states for all VLANs. Although the CSS maintains a spanning-tree instance for each VLAN as well, the CSS uses the standard 01-80-C2-00-00-00 destination MAC address for all BPDUs (tagged or untagged). When you connect a Cisco Catalyst switch to a CSS over an 802.1Q trunk, the result is that neither switch recognizes the other's BPDUs, and both assume root status. If a spanning-tree loop is detected, the Catalyst switch goes into blocking mode on one of its looped ports.

This chapter contains the following major sections:

For details about configuring spanning-tree bridging parameter for an Ethernet interface or for a trunked Ethernet interface and VLAN pair, refer to Chapter 1, Configuring Interfaces and Circuits.

CSS Spanning-Tree Bridging Quick Start

Table 2-1 provides a quick overview of the steps required to globally configure spanning-tree bridging for the CSS. 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 2-1.

Table 2-1 Spanning-Tree Bridging Configuration Quick Start

Task and Command Example
1. Set the bridge filtering database aging time, in seconds, for the CSS. `(config)# bridge aging-time 600`
2. Set the bridge forward delay time, in seconds, that the bridge uses when acting as the root. `(config)# bridge forward-time 9`
3. Set the bridge hello time interval, in seconds, that the bridge waits before sending a hello packet. `(config)# bridge hello-time 9`

Table 2-1 Spanning-Tree Bridging Configuration Quick Start (continued)

Task and Command Example
4. Set the bridge spanning-tree maximum age, in seconds. `(config)# bridge max-age 21`
5. Set the priority that the bridge spanning tree uses to choose the root bridge in the network. `(config)# bridge priority 1700`
6. (Recommended) Display bridge forwarding information. `(config)# show bridge status`

The following running-configuration example shows the results of entering the commands in Table 2-1.

```
!************************** GLOBAL ***************************
  bridge aging-time 600
  bridge forward-time 9
  bridge hello-time 9
  bridge max-age 21
  bridge priority 1700
```

Configuring Spanning-Tree Bridge Aging-Time

The aging time is the timeout period, in seconds, for aging out dynamically learned forwarding information. By default, the bridge filtering database aging time for the CSS is 300 seconds. To set the bridge filtering database aging time for the CSS., use the **bridge aging-time** command. Enter an integer from 10 to 1000000.

To set the bridge aging time to 600, enter:

```
(config)# bridge aging-time 600
```

To restore the default aging time of 300, enter:

```
(config)# no bridge aging-time
```

Configuring Spanning-Tree Bridge Forward-Time

The forward time is the delay time, in seconds, that all bridges use for forward delay when this bridge is acting as the root. By default, the bridge forward delay time is 4 seconds. Use the **bridge forward-time** command to set the bridge forward delay time. Enter an integer from 4 to 30.

To set the bridge forward time to 9, enter:

```
(config)# bridge forward-time 9
```

To restore the default delay time of 4, enter:

```
(config)# no bridge forward-time
```

Configuring Spanning-Tree Bridge Hello-Time

The hello time is the time, in seconds, that all bridges wait before sending a hello packet (when the bridge acts as the root). By default, the bridge hello time interval is 1 second. Use the **bridge hello-time** command to set the bridge hello time interval. Enter an integer from 1 to 10.

To set the bridge hello time to 9, enter:

```
(config)# bridge hello-time 9
```

To restore the default hello time interval of 1, enter:

```
(config)# no bridge hello-time
```

Configuring Spanning-Tree Bridge Max-Age

The maximum age is the time, in seconds, that protocol information received on a port is stored by the CSS (when a bridge acts as the root). By default, the bridge spanning-tree maximum age is 6 seconds. Use the **bridge max-age** command to set the bridge spanning-tree maximum age. Enter an integer from 6 to 40.

Note Ensure the bridge maximum age is greater than or equal to 2 times (bridge hello-time + 1 second) and less than or equal to 2 times (bridge forward-time - 1 second).

To set the bridge maximum age to 21, enter:

```
(config)# bridge max-age 21
```

To restore the default maximum age of 6, enter:

```
(config)# no bridge max-age
```

Configuring Spanning-Tree Bridge Priority

In spanning tree, the 2-octet field is prepended to the 6-octet MAC address to form an 8-octet bridge identifier. The device with the lowest bridge identifier is considered the highest priority bridge and becomes the root bridge. By default, the bridge priority is set to 32768. Use the **bridge priority** command to set the priority that the bridge spanning tree uses to choose the root bridge in the network. The range for bridge priority is 0 to 65535.

For example:

```
(config)# bridge priority 1700
```

To restore the bridge priority to the default of 32768, enter:

```
(config)# no bridge priority
```

Disabling Bridge Spanning-Tree

Spanning-tree bridging is enabled by default. When you disable spanning-tree bridging, the CSS drops those bridge protocol data units (BPDUs) that it recognizes as BPDUs, but forwards the Cisco Systems 802.1Q BPDUs (tagged with the proprietary 01-00-0c-cc-cc-cc-cd destination MAC address) on an 802.1Q VLAN trunk. The CSS can still operate in an 802.1Q spanning-tree environment as long as you do not require that the CSS put any of its ports into a blocking state.

Caution Disabling spanning-tree bridging may make your network susceptible to packet storms.

To disable spanning-tree bridging, enter:

```
(config)# bridge spanning-tree disable
```

To reenable spanning-tree bridging, enter:

```
(config)# bridge spanning-tree enable
```

Showing Bridge Configurations

Use the **show bridge forwarding** command to display bridge forwarding information. Table 2-2 describes the fields in the **show bridge forwarding** command output.

Table 2-2 Field Descriptions for the show bridge forwarding Command

Field	Description
VLAN	The bridge interface virtual LAN number
MAC Address	The MAC address for the entries
Port Number	The port number used for bridge forwarding

Use the **show bridge status** command to display bridge status information. Table 2-3 describes the fields in the **show bridge status** output.

Table 2-3 Field Descriptions for the show bridge status Command

Field	Description
STP State	The state of the Spanning-Tree Protocol: Enabled or Disabled.
Root Max Age	The timeout period, in seconds, during which the host times out root information.
Root Hello Time	The interval, in seconds, during which the root bridge broadcasts its hello message to other devices.
Root Fwd Delay	The delay time, in seconds, that the root bridge uses for forward delay.
Designated Root	The bridge ID for the designated root.

Table 2-3 Field Descriptions for the show bridge status Command (continued)

Field	Description
Bridge ID	The bridge ID of the bridge.
Port	The port ID.
State	The state of the port. The possible states are as follows: • Block - The blocking state. A port enters the blocking state after CSS initialization. The port does not participate in frame forwarding. • Listen - The listening state. This state is the first transitional state a port enters after the blocking state. The port enters this state when STP determines that the port should participate in frame forwarding. • Learn - The learning state. The port enters the learning state from the listening state. The port in the learning state prepares to participate in frame forwarding. • Forward - The forwarding state. The port enters the forwarding state from the learning state. A port in the forwarding state forwards frames. • Disabled - The disabled state. A port in the disabled state does not participate in frame forwarding or the Spanning-Tree Protocol. A port in the disabled state is non operational.
Designated Bridge	The bridge ID for the designated bridge.
Designated Root	The bridge ID for the designated root.
Root Cost	The cost of the root.
Port Cost	The cost of the port.
Desg Port	Designated port.

Configuring Open Shortest Path First

This chapter provides configuration and viewing information for the Open Shortest Path First (OSPF) protocol. Information in this chapter applies to all CSS models, except where noted.

Note The CSS supports OSPF Version 2, as defined in RFC 2178. For detailed information about OSPF MIB objects, refer to RFC 1850.

This chapter contains the following major sections:

OSPF Overview

OSPF is a link-state routing protocol that:

- Provides network topology discovery within a group of routers and networks called an autonomous system (AS)
- Calculates the shortest path to destinations within the AS

As a link-state protocol, OSPF routers flood any change in routing information throughout the network. This action differs from a distance vector protocol, such as RIP, which periodically exchanges routing information only with neighboring devices.

Within an AS, each OSPF router builds and synchronizes a database of the AS network topology. The routers synchronize their databases by requesting information from other AS routers. Each router sends its information as link-state advertisements (LSAs) that include information about the state of each router and link in the AS. A link is an interface on the router. The state of the link is the description of the interface, including the router's IP address and subnet mask, and its relationship to the neighboring router.

Then, the router uses its database and the Shortest Path First (SPF) algorithm to calculate the shortest path to every destination in the AS and stores this information in a dynamic table. When changes occur, the router calculates new paths.

The CSS, operating as an OSPF router, provides:

- Intra-area route support for routing in a single area between other OSPF routers
- Inter-area route support for routing between multiple OSPF areas
- Route summarization between areas as an Area Border Router (ABR)
- Stub area and AS boundary router support
- Redistribution of local, RIP, static, and firewall routes into an OSPF domain
- Advertisement of VIP addresses for content as AS external routes
- Simple authentication

This section includes the following topics:

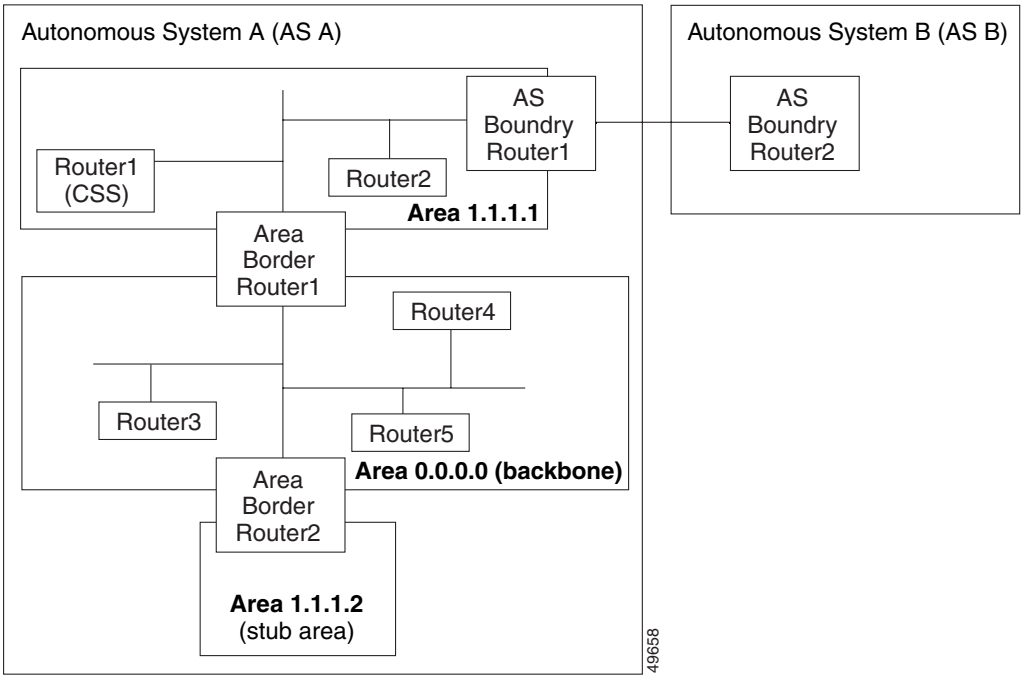
OSPF Routing Hierarchy

The OSPF routing hierarchy includes the following functions:

- Autonomous systems
- Areas, including the backbone and stub areas
- Area Border Routers (ABRs)
- Autonomous System Boundary Routers (ASBRs)

Figure 3-1 illustrates an OSPF network topology.

Figure 3-1 Basic OSPF Network Topology



Autonomous System

The autonomous system (AS) is a collection of networks, under the same administrative control, that share the same routing information with each other. An AS is also referred to as a routing domain. Figure 3-1 shows two ASs: AS A and AS B. An AS can consist of one or more OSPF areas.

Areas

Areas allow the subdivision of an AS into smaller, more manageable networks or sets of adjacent networks. As shown in Figure 3-1, AS A consists of three areas: area 0.0.0.0, area 1.1.1.1, and area 1.1.1.2.

OSPF hides the topology of an area from the rest of the AS. An area's network topology is visible only to routers inside that area; the network topology is not visible to routers outside the area. When OSPF routing is within an area, this is called intra-area routing. This routing limits the amount of link-state information flooding onto the network, thereby reducing routing traffic. OSPF routing also reduces the size of the topology information in each router, which conserves processing and memory requirements in each router.

Conversely, the routers within an area cannot see detailed network structures outside the area. Because of this restriction of topological information, you can control traffic flow between areas and reduce routing traffic when the entire autonomous system is a single routing domain.

Backbone Area

A backbone area is responsible for distributing routing information between the areas of an autonomous system. When OSPF routing occurs outside of an area, this is called inter-area routing.

The backbone itself has all the properties of an area. It consists of ABRs, and routers and networks only on the backbone. As shown in Figure 3-1, area 0.0.0.0 is an OSPF backbone area. Note that a designated OSPF backbone area has a reserved ID of 0.0.0.0.

Area Border Routers

ABRs have multiple interfaces that connect directly to networks in two or more areas. An ABR runs a separate copy of the OSPF algorithm and maintains separate routing data for each area that is connected to it, including the backbone area. ABRs also send configuration summaries for their attached areas to the backbone area, which distributes this information to other OSPF areas in the autonomous system. In Figure 3-1, there are two ABRs. ABR 1 interfaces area 1.1.1.1 to the backbone area. ABR 2 interfaces the backbone area to area 1.1.1.2, a stub area.

Note ABRs are always backbone routers. You must configure ABRs to the backbone area.

Stub Area

A stub area is an area that does not accept or distribute detailed network information external to the area. A stub area has only one router that interfaces the area to the rest of the AS. The ABR attached to the stub area advertises a single default external route into the area. Routers within a stub area use this route for destinations outside the autonomous system, as well as for inter-area routes. This relationship conserves LSA database space that would otherwise be used to store external LSAs flooded into the area. As shown in Figure 3-1, area 1.1.1.2 is a stub area that is reached only through ABR 2.

Autonomous System Boundary Routers

ASBRs provide connectivity from one autonomous system to another system. ASBRs exchange their autonomous system routing information with boundary routers in other autonomous systems. Every router inside an autonomous system knows how to reach the boundary routers for its autonomous system.

ASBRs can import external routing information from other protocols like RIP and redistribute them as AS-external LSAs to the OSPF network. If the CSS is an ASBR, you can configure it to advertise VIP addresses for content as AS external routes. In this way, ASBRs flood information about external networks to routers within the OSPF network.

ASBR routes can be advertised as type1 or type2 ASE. The difference between type1 and type2 is how the cost is calculated. For a type2 ASE, only the external cost (metric) is used when comparing multiple paths to the same destination. For type1 ASE, the combination of the external cost and the cost to reach the ASBR is used.

Link-State Databases

OSPF routers advertise routes using LSAs. The link-state database stores the LSAs from routers throughout the area. The advertisements depict the topology of the autonomous system. They could include:

- Router links that describe the state and cost of each router's interface to an area
- Network links from the designated router (see the "Setting the Priority of the CSS" section) that describe all routes on a segment for multi-access segments with more than one attached router
- Summarized links from ABRs that describe networks in the AS but outside an area
- External links from ASBRs that describe destinations external to the AS

All routers that are connected to an area maintain identical routing databases about the area. Routers that are connected to multiple areas maintain a separate routing database for each attached area.

Instead of each router sending routing information to every other router on the network, OSPF routers establish adjacencies among neighboring routers. When the link-state databases of two neighboring routers are synchronized, they are considered adjacent.

OSPF routers collect raw topological data from the LSAs that they receive. Each router then prunes this data down to a tree of the shortest network paths centered on itself. The router examines the total cost to reach each router or network node in its domain. By discarding all but the lowest-cost path to each destination, the router builds a shortest-path tree to each destination, which it uses until the network topology changes. It is possible to have multiple lowest-cost paths to a destination.

CSS OSPF Configuration Quick Start

This section includes the following topics:

Global OSPF Configuration Quick Start

To perform the global OSPF configuration for the CSS, see the steps in Table 3-1. In the most basic global configuration, where the CSS functions as a router in the OSPF backbone area, you need to perform only steps 1 and 2 to:

- Define the CSS router ID
- Enable OSPF

Optionally, you can define the CSS:

- In an area other than the backbone, including a stub area.
- As an ABR, by configuring route summarization.
- As an ASBR, to advertise non-OSPF routes through OSPF, as AS-external routes such as static and RIP routes. You could also advertise VIP addresses for content as AS external routes.

After performing the global OSPF configuration, you must configure an OSPF IP interface (see the "OSPF IP Interface Configuration Quick Start" section) before the CSS can participate in OSPF routing. For more information on configuring global OSPF parameters, see the "Configuring OSPF on the CSS" section.

Table 3-1 Global OSPF Configuration Quick Start

Task and Command Example
1. Configure the area router ID for the CSS in global configuration mode. In this example, the CSS router ID is 121.23.21.1. `(config)` **`ospf router-id 121.23.21.1`**
2. (Optional) If the CSS area is other than the backbone area, enter the area ID for the CSS. In this example, the area ID is 1.1.1.1. `(config)` **`ospf area 1.1.1.1`** The default ID is 0.0.0.0 for the backbone area. To define a stub area, enter the stub option after the area ID.
3. (Optional) If you want the CSS to advertise external routes, define the CSS as an AS boundary router. For example: `(config)` **`ospf as-boundary`**
4. (Optional) If the CSS is an ABR, you can advertise VIP addresses for content as OSPF ASE routes. To advertise the VIP address 192.168.4.15 with a default cost of 1 and the default type of ASE type2, enter: `(config)` **`ospf advertise 192.168.4.15 255.255.255.255`**
5. (Optional) To advertise routes other than OSPF, such as a firewall, local, RIP or static route, configure OSPF to redistribute routes from the specific protocol. To advertise static routes through OSPF with a default cost of 1 and default type of ASE type2, enter: `(config)` **`ospf redistribute static`**
6. Enable OSPF on the CSS. `(config)` **`ospf enable`**

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

```
!************************** GLOBAL ***************************

  ospf router-id 121.23.21.1
  ospf area 1.1.1.1
  ospf as-boundary
  ospf advertise 192.168.4.15 255.255.255.255
  ospf redistribute static
  ospf enable
```

OSPF IP Interface Configuration Quick Start

To configure OSPF on a CSS IP interface, see the steps in Table 3-2. In the most basic IP interface configuration, you need to perform only steps 1 through 4, and step 7 to:

- Assign OSPF to the IP interface
- Associate OSPF with the globally defined area, if this is an area other than the backbone area (0.0.0.0)
- Enable OSPF on the interface

This configuration example assumes you will accept the default OSPF configuration settings for the interface, except the router priority. The interface OSPF configuration settings include:

- Intervals for the hello packet, LSA retransmission, and link-state update packet
- Authentication password
- CSS router priority
- Interface cost

For more information on configuring these OSPF IP interface settings, see the "Configuring OSPF on a CSS IP Interface"section.

Table 3-2 Configuration Quick Start for OSPF on a CSS Interface

Task and Command Example
1. Access global configuration mode. Enter: `# config`

2. Access the circuit configuration mode for a preconfigured circuit on which you want to create the IP interface. For example, if circuit VLAN6 already exists, enter:
`(config)# circuit VLAN6`
`(config-circuit[VLAN6])#`

Note Refer to Chapter 1, Configuring Interfaces and Circuits for information on how to configure the CSS interfaces and circuits and the bridge interfaces to VLANs.

Table 3-2 Configuration Quick Start for OSPF on a CSS Interface (continued)

Task and Command Example

3. Create the IP interface to the circuit. To create an IP address of 3.1.2.2 with a subnet mask of /24, enter:
`(config-circuit[VLAN6])# ip address 3.1.2.2/24`
`Create ip interface <3.1.2.2>, [y/n]: y`

4. Configure the IP interface as an OSPF interface. Enter: `(config-circuit-ip[VLAN6-3.1.2.2])# ospf`
5. (Optional) If the globally configured area is other than the backbone area, enter the configured area ID. In this example, the globally configured area ID is 1.1.1.1. `(config-circuit-ip[VLAN6-3.1.2.2]) ospf area 1.1.1.1`
6. (Optional) With a default setting of 1, the CSS is set to a priority that allows it to become the designated router. If you do not want the CSS to become the designated router, you can change its priority or disable it from eligibility. For example, if you want the CSS to be ineligible to become a designated router, enter: `(config-circuit-ip[VLAN6-3.1.2.2])# ospf priority 0` For more information on designated routers, see the "Setting the Priority of the CSS" section.
7. Enable OSPF on the interface. Enter: `(config-circuit-ip[VLAN6-3.1.2.2])# ospf enable`

The following running-configuration example shows the results of entering the commands in Table 3-2.

```
!************************* CIRCUIT ************************
circuit VLAN6

  ip address 3.1.2.2 255.255.255.0
    ospf
    ospf area 1.1.1.1
    ospf priority 0
```

Verifying Your Configuration

To verify the OSPF global and interface configurations, use the **show ospf** command and its options. For example:

- To show the OSPF global configuration, use the **show ospf global** command. For example:

  ```
  # show ospf global
  ```

  If the Admin Status field is disabled, use the **ospf enable** command to enable OSPF.

- To show the route redistribution policy into OSPF, use the **show ospf redistribute** command. To show the configured static route redistribution policy, enter:

  ```
  # show ospf redistribute
  ```

- To show the VIP addresses advertised as ASE routes, use the **show ospf advertise** command. For example:

  ```
  # show ospf advertise
  ```

- To view the CSS IP interface configuration, use the **show ospf interfaces** command. For example:

  ```
  # show ospf interfaces
  ```

Configuring OSPF on the CSS

This section includes the following topics:

Configuring the OSPF Router ID

Before you enable OSPF on the CSS, configure the router ID. Assigning a router ID to the CSS uniquely identifies it to other routers within the autonomous system. In addition, in the case of a priority tie when determining which router is the designated router, the ID serves as a tie-breaker in the designated router election. For more information on designated routers, see the "Setting the Priority of the CSS" section.

Use the **ospf router-id** command to configure the OSPF router ID for the CSS. A router ID is a 32-bit number in dotted-decimal notation.

To assign the router ID of 121.23.21.1 to the CSS, enter:

```
(config)# ospf router-id 121.23.21.1
```

Note If OSPF is globally enabled, use the **no** form of the **ospf enable** command to disable OSPF and change the router ID.

To delete the router ID on the CSS, disable OSPF and enter:

```
(config)# no ospf router-id
```

Enabling OSPF

After you assign the router ID to the CSS, globally enable OSPF on the CSS. Use the **ospf enable** command to enable OSPF. For example:

```
(config)# ospf enable
```

To disable OSPF, enter:

```
(config)# no ospf enable
```

Configuring an Area

By default, the CSS is configured to the backbone area automatically. The backbone area has a reserved ID of 0.0.0.0. If the CSS is part of an area other than the backbone area, assign the CSS to that area.

Use the **ospf area** command to assign an area. Enter the ID in dotted-decimal notation (for example, 0.0.0.1). Although an area ID has the same form as an IP address, the area ID address space is its own distinct address space.

For example, if the CSS is in area 0.0.0.1, enter:

```
(config)# ospf area 0.0.0.1
```

If the CSS is in a stub area, include the **stub** option.

For example, if area 0.0.0.1 is a stub area, enter:

```
(config)# ospf area 0.0.0.1 stub
```

Optionally, for a stub area you can also:

- Set a metric for the default route advertised in the stub area.
- Propagate summary LSAs into the stub area.

To set a metric for the default route advertised in the stub area, include the **default-metric** option. By default, the metric equals the smallest metric among the interfaces to other areas. You can assign an integer from 1 to 16777215.

For example, to assign a metric of 200, enter:

```
(config)# ospf area 0.0.0.1 stub default-metric 200
```

To propagate summary LSAs in the stub area, include the **send-summaries** option. For example:

```
(config)# ospf area 0.0.0.1 stub send-summaries
```

Removing an Area

To remove an OSPF area, disable OSPF, then use the **no** form of the **ospf area** command. For example:

```
(config)# no ospf enable
(config)# no ospf area 0.0.0.1
```

Configuring Equal-Cost Routes

By default, the OSPF CSS is configured to use 15 equal-cost routes. Use the **ospf equal-cost** command to change the number of routes. Enter a number from 1 to 15.

To configure 10 equal-cost routes for use by the CSS, enter:

```
(config)# ospf equal-cost 10
```

To reset the equal-cost routes to its default value of 15, enter:

```
(config)# no ospf equal-cost
```

Configuring Summarized Routes at an ABR

If the CSS is an ABR, you can configure it to advertise a single summary route or network ranges that cover all the individual networks within the specified range. This summarization helps control routing table sizes and prevents the constant changing of routes whenever an interface within an area comes online or goes offline. These route changes do not cause route changes in backbone ABRs and other area routers.

Use the **ospf range** command to specify the IP address range to summarize routes at the ABR. This summarization applies to inter-area paths that are paths to destinations in other OSPF areas. You can also determine whether you want to advertise this range. Disable OSPF before you enter the **ospf range** command.

Define an address range by specifying an IP address and subnet mask that represents networks in the area being summarized. Enter the IP address and subnet mask in dotted-decimal notation (for example, 192.168.128.0 255.255.224.0). You can also enter the mask in CIDR bit-count notation format (for example, /24).

To configure the CSS as an ABR with an area ID of 0.1.0.1 with a collection of destinations between 192.168.0.0 and 192.168.255.255, enter:

```
(config)# no ospf enable
(config)# ospf range 0.1.0.1 192.168.0.0 255.255.0.0
```

To remove the range, enter:

```
(config)# no ospf range 0.1.0.1 192.168.0.0 255.255.0.0
```

By default, the ABR advertises this range. If you want to hide the range from the rest of the AS, include the **block** option. For example:

```
(config)# ospf range 0.1.0.1 192.168.0.0 255.255.0.0 block
```

Configuring the CSS as an Autonomous System Boundary Router

If you want the CSS to be an ASBR that exchanges routing information with routers belonging to other autonomous systems, use the **ospf as-boundary** command. Disable OSPF before you enter the **ospf as-boundary** command.

For example:

```
(config)# no ospf enable
(config)# ospf as-boundary
```

To remove the CSS as an AS boundary router, enter:

```
(config)# no ospf as-boundary
```

To advertise a route as OSPF ASE through all OSPF interfaces or generate a default route, see the following sections.

- Advertising a Route as an OSPF ASE Route
- Advertising a Default ASE Route
- Advertising Other Routes Through OSPF

Advertising a Route as an OSPF ASE Route

The CSS OSPF functionality examines configuration parameters (such as service configurations in content rules, keepalive behavior, VIP redundancy configurations, and whether services are active or suspended) to make accurate advertisement decisions on VIPs.

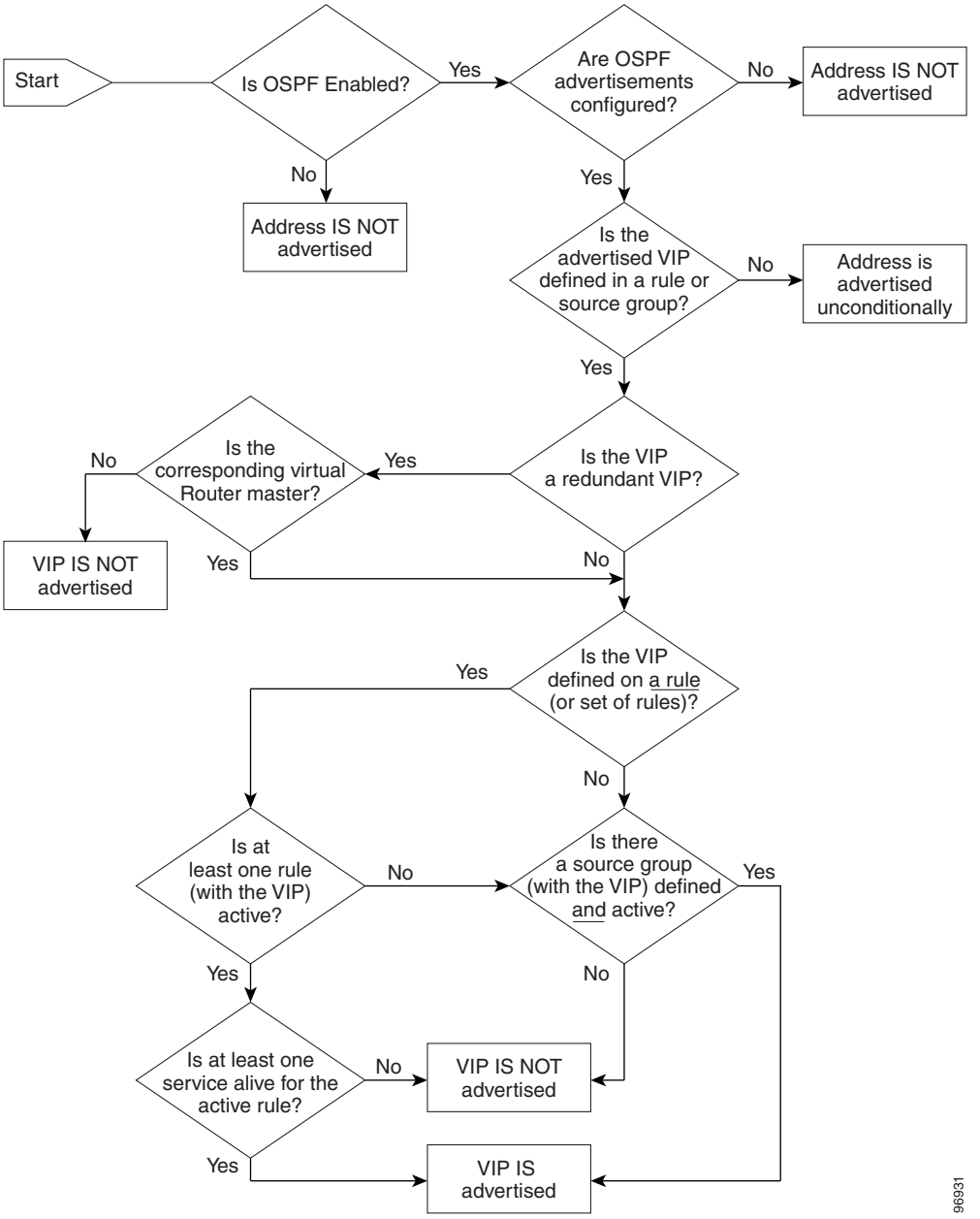
Specified routes related to VIPs are only advertised if both of the following conditions are true:

- At least one of the related VIPs in a content rule or source group is active.
- At least one service related to an active VIP is available on a content rule.

If you configured the CSS for box-to-box redundancy, be aware that only the master CSS (not the backup CSS) advertises the VIP.

We recommend that you use the /32 prefix in the **ospf advertise** command to specify VIPs individually. Specifying entire subnets does not enable the CSS to make proper decisions on advertising the VIPs. The advertisement must match or fit entirely within a VIP range to make proper decisions. If the OSPF advertise IP address range and the VIP range overlap, or the OSPF advertise range encapsulates (that is, is larger than) or doesn't match the VIP range, then the route is advertised unconditionally.

The following flow chart shows the steps required for OSPF to advertise an IP address. If the IP address is a VIP, the flowchart shows the conditions that must be met for OSPF to advertise the VIP.


Start
Is OSPF Enabled?
Yes
Are OSPF advertisements configured?
No
Address IS NOT advertised
No
Address IS NOT advertised
Yes
Is the advertised VIP defined in a rule or source group?
No
Address is advertised unconditionally
Yes
Is the VIP a redundant VIP?
Yes
Is the corresponding virtual Router master?
No
VIP IS NOT advertised
Yes
No
Is the VIP defined on a rule (or set of rules)?
Yes
No
Is at least one rule (with the VIP) active?
No
Is there a source group (with the VIP) defined and active?
Yes
Yes
No
Is at least one service alive for the active rule?
No
VIP IS NOT advertised
Yes
VIP IS advertised
96931

The ASBR can perform external route summarization to consolidate multiple routes into a single advertisement. For a CSS, this consolidation is useful when you want to advertise VIP addresses for content as OSPF AS external (ASE) through all OSPF interfaces. Use the **ospf advertise** command to advertise a route as OSPF ASE through all OSPF interfaces. To stop the advertisement of the route, use the **no** form of the **ospf advertise** command (as described later in this section).

Note When using OSPF to advertise a VIP address, do not configure this address on a content rule as a single VIP address when another content rule includes it within its VIP address range. If you do, OSPF may make erroneous advertisement decisions or some rules may appear to have the wrong VIP redundancy state associated with them.

First, before you enter the **ospf advertise** command, configure the CSS as an ASBR. For more information, see the "Configuring the CSS as an Autonomous System Boundary Router" section.

Define an address range for the **ospf advertise** command by specifying an IP address and subnet mask that represents networks in the area being summarized. Enter the IP address and subnet mask in dotted-decimal notation (for example, 192.168.128.0 255.255.224.0). You can also enter the mask in CIDR bit-count notation format (for example, /24).

For example, to advertise VIP addresses from 192.168.44.0 to 192.168.44.255, define the range by entering the IP address and subnet mask of 192.168.44.0 255.255.255.0:

```
(config)# ospf advertise 192.168.44.0 255.255.255.0
```

We recommend that you use the /32 prefix in the **ospf advertise** command to specify VIPs individually. Specifying entire subnets does not enable the CSS to make proper decisions on advertising the VIPs. The advertisement must match or fit entirely within a VIP range to make proper decisions. If the OSPF advertise IP address range and the VIP range overlap, or the OSPF advertise range encapsulates (that is, is larger than) or doesn't match the VIP range, then the route is advertised unconditionally.

Optionally, you can define any of the following:

- The network cost for the route by including the **metric** option. Enter a number from 1 to 16777215. The default is 1.
- A 32-bit tag value to advertise each external route by including the **tag** option. The 32-bit tag value is not used by the OSPF protocol itself. You can use the tag value to communicate information between ASBRs.
- The advertised routes as ASE type1 by including the **type1** option. By default, the type is ASE type2. The difference between type1 and type2 is how the cost is calculated. For a type2 ASE, only the external cost (metric) is used when comparing multiple paths to the same destination. For type1 ASE, the combination of the external cost and the cost to reach the ASBR is used.

For example:

```
(config)# ospf advertise 193.23.44.0 255.255.255.0 metric 3 type1
```

To stop advertising of the route as OSPF ASE through all OSPF interfaces, enter:

```
(config)# no ospf advertise 193.23.44.255.255.255.0
```

The following running configuration example illustrates the **ospf advertise** command for OSPF advertising of VIP addresses and an IP address. Comments are preceded by an exclamation point (!).

```
!************************** GLOBAL ***************************
ospf enable

ospf advertise 1.1.1.10
!advertise redundant VIP
ospf advertise 2.1.1.1
!advertise IP address of service s1
ospf advertise 1.1.1.100
!advertise IP address of critical service c100
ospf advertise 99.99.99.99
!advertise simple IP address, not tied to anything
record

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

ip address 1.1.1.200 255.0.0.0
ip virtual-router 1
ip redundant-vip 1 1.1.1.10
!redundant VIP
ip critical-service 1 c100
```

```
!************************** SERVICE **************************
service c100
ip address 1.1.1.100
!IP address for critical service
active

service s1
ip address 2.1.1.1
!IP address for service s1
keepalive method get
keepalive type http
active

service s2
ip address 2.1.1.2
keepalive method get
keepalive type http
active

!************************** OWNER **************************
owner admin1

content r1
add service s1
add service s2
vip address 1.1.1.10
!redundant VIP equals content VIP

active
```

Advertising a Default ASE Route

Routers use default routes when no additional routes exist to a particular AS external destination. By default, an ASBR does not generate a default route into the OSPF routing domain. Use the **ospf default** command to force the CSS to generate a default ASE route and advertise the route through OSPF.

Before you enter the **ospf default** command, configure the CSS as an ASBR. For more information, see the "Configuring the CSS as an Autonomous System Boundary Router" section.

For example:

```
(config)# ospf default
```

Optionally, you can define any of the following:

- The network cost for an OSPF default route by including the **metric** option. If a default route metric is defined, the router advertises itself as the default router to the area. Enter a number from 1 to 16,777,215. The default is 1.
- A 32-bit tag value to advertise each external route by including the **tag** option. The 32-bit tag value is not used by the OSPF protocol itself. You can use the tag value to communicate information between ASBRs.
- The advertised routes as ASE type1 by including the **type1** option. By default, the type is ASE type2. The difference between type1 and type2 is how the cost is calculated. For a type2 ASE, only the external cost (metric) is used when comparing multiple paths to the same destination. For type 1 ASE, the combination of the external cost and the cost to reach the ASBR is used.

For example:

```
(config)# ospf default metric 10 type1
```

To stop advertising the default ASE routes originated through OSPF, enter:

```
(config)# no ospf default
```

Advertising Other Routes Through OSPF

To advertise routes from other protocols, such as firewall, local, RIP, and static routes through OSPF, use the **ospf redistribute** command. Redistribution of these routes makes them OSPF external routes.

To redistribute routes from other protocols, include one of the following options:

- **firewall** - Advertises firewall routes through OSPF
- **local** - Advertises local routes (interfaces *not* running OSPF)
- **rip** - Advertises RIP routes through OSPF
- **static** - Advertises static routes configured for the Ethernet interface ports. The **ospf redistribute static** command does not advertise static routes configured for the Ethernet management port.

To advertise a firewall route, enter:

```
(config)# ospf redistribute firewall
```

Optionally, you can define any of the following:

- The network cost for the route by including the **metric** option. Enter a number from 1 to 16,777,215. The default is 1.
- A 32-bit tag value to advertise each external route by including the **tag** option. The 32-bit tag value is not used by the OSPF protocol itself. You can use the tag value to communicate information between AS boundary routers.
- The advertised routes as ASE type1 by including the **type1** option. By default, the type is ASE type2. The difference between type1 and type2 is how the cost is calculated. For a type2 ASE, only the external cost (metric) is considered when comparing multiple paths to the same destination. For type1 ASE, the combination of the external cost and the cost to reach the ASBR is used.

For example:

```
(config)# ospf redistribute rip metric 3 type1
```

To stop advertising the RIP routes via OSPF, enter:

```
(config)# no ospf redistribute rip
```

Configuring OSPF on a CSS IP Interface

When you configure a CSS IP interface as an OSPF interface, you define its behavior and role within the OSPF routing domain. This section includes the following topics:

- Configuring the CSS IP Interface as an OSPF Interface
- Assigning an OSPF Area to the Interface
- Enabling OSPF on the Interface
- Configuring the Interface Attributes

Configuring the CSS IP Interface as an OSPF Interface

An OSPF interface is an IP interface that you configure to send and receive OSPF traffic. To configure the CSS IP interface as an OSPF interface, use the **ospf** command.

Note You must enter the **ospf** command before the **ospf enable** command can take effect.

To configure the CSS IP interface as an OSPF interface:

1. Access the circuit configuration mode for the preconfigured circuit on which you want to create the IP interface. For example, if circuit VLAN6 already exists, enter:

```
(config)# circuit VLAN6
(config-circuit[VLAN6])#
```

Note Refer to Chapter 1, Configuring Interfaces and Circuits for information on how to configure the CSS interfaces and circuits, and bridge interfaces to VLANs.

2. Create the IP interface to the circuit. To create an IP address of 3.1.2.2, enter:

```
(config-circuit[VLAN6])# ip address 3.1.2.2/24
Create ip interface <3.1.2.2>, [y/n]:y
```

3. Configure this circuit as an OSPF circuit. Enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf
```

Assigning an OSPF Area to the Interface

After you configure the IP interface as an OSPF interface, assign it to the area that you globally configured to the CSS. The default area is the backbone area with the ID of 0.0.0.0. If the area is other than the backbone, use the **ospf area** command to assign the interface to an OSPF area. For example, if the area is 0.0.0.1, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf area 0.0.0.1
```

To reset the interface to the default backbone area, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf area
```

Enabling OSPF on the Interface

If you need to configure the interface attributes as described in the "Configuring the Interface Attributes" section, do not enable OSPF on the IP interface until you finish configuring the attributes.

By default, OSPF is disabled on an IP interface. Use the **ospf enable** command to enable OSPF on the IP interface. For example:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf enable
```

To disable OSPF on the interface, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf enable
```

Configuring the Interface Attributes

The OSPF interface attributes are set to a series of default values. You can elect to use these values for the CSS IP interface or configure your own settings. This section includes the following topics:

- Setting the Cost
- Setting the Dead Router Interval
- Setting the Hello Packet Interval
- Setting the Password
- Setting the Poll Interval

- Setting the Priority of the CSS
- Setting the Retransmission Interval
- Setting the Transit-Link Delay

Setting the Cost

To set the cost for sending a data packet on this interface, use the **ospf cost** command. The cost for the interface is a number from 0 to 65535. The default value of the cost for a given type of circuit is 108/interface speed. For a Gigabit Ethernet interface, the value is 1. For a 10/100-Mbps Fast Ethernet interface, the value is 10.

For example, to set a cost of 25, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf cost 25
```

To reset the packet cost for the interface to the default value, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf cost
```

Setting the Dead Router Interval

The interface declares a neighbor router is dead if the interface does not receive hello packets from the router before the dead interval expires. Use the **ospf dead** command to set the dead router interval for an interface. The dead router interval is in seconds. This value must be a multiple of the hello interval, and the value must be the same for all routers attached to a common network. Enter a number from 1 to 2,147,483,647. The default is 40.

For example, to set the dead router interval to 100 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf dead 100
```

To reset the dead router interval to its default of 40 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf dead
```

Setting the Hello Packet Interval

Router interfaces periodically transmit hello packets to identify and maintain communications with their neighbors. When a router detects its own address in another router's hello packet, the two routers establish two-way communications as neighbors.

The hello interval is the length of time, in seconds, between hello packets that the interface sends to its neighbor routers. The hello interval must be the same value for all routers attached to a common network. Use the **ospf hello** command to set the hello interval for the IP interface. Enter an integer from 1 to 65535. The default is 10 seconds.

To set a hello interval of 25 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf hello 25
```

To reset the hello interval to the default value of 10 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf hello
```

Setting the Password

All OSPF protocol exchanges can be authenticated to ensure only known, trusted routers participate in routing updates. The OSPF password is used for authentication of all OSPF protocol exchanges.

Use the **ospf password** command to set the password for an interface. This password must be the same for all routers attached to a common network. Enter a quoted text string with a maximum of eight characters.

For example, to set the password of *quota*, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf password "quota"
```

To remove the OSPF password from the interface, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf password
```

Setting the Poll Interval

The poll interval is the length of time, in seconds, between the transmittal of hello packets by the CSS to an assumed inactive neighbor router in a non-broadcast, multi-access network. Use the **ospf poll** command to set the poll interval for the interface. The poll interval should be a value that is greater than the hello time interval. Enter a number from 1 to 2,147,483,647. The default is 120 seconds.

Note The **ospf poll** command has no effect when you operate the CSS over a broadcast LAN (that is, an Ethernet network).

For example, to set the poll interval to 200 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf poll 200
```

To reset the poll interval to the default value of 120 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf poll
```

Setting the Priority of the CSS

To avoid the need for each router on a LAN to talk to every router on a network that has more than two attached routers, one router is elected as the designated router. Designated routers advertise network link states for attached network segments. An LSA lists all routers that are connected to a segment.

The priority determines which router is the designated router. The router with the highest priority becomes the designated router. In case of a tie, routers use their router ID as a tie breaker.

Use the **ospf priority** command to set the router priority for the interface. The priority of the interface is an integer from 0 to 255. The default is 1, which is the highest router priority. A value of 0 signifies that the CSS is not eligible to become the designated router on a particular network.

If a designated router exists on the network, it remains the designated router regardless of its router priority.

To make the interface ineligible to become a designated router, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf priority 0
```

To reset the router priority to the default value of 1, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf priority
```

Setting the Retransmission Interval

The retransmission interval is the number of seconds between link-state advertisement retransmissions for adjacencies belonging to an interface. OSPF creates adjacencies between neighboring routers for the purpose of exchanging routing information. The CSS also uses the interval when retransmitting database descriptions and link-state request packets.

Use the **ospf retransmit** command to set the retransmit interval for the interface. Enter a number from 1 to 3600 seconds (1 hour). The default is 5 seconds.

To set the retransmission interval to 10 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf retransmit 10
```

To reset the retransmit interval to the default value of 5 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf retransmit
```

Setting the Transit-Link Delay

Transit delay is the estimated number of seconds the CSS waits to transmit a link-state update packet over the OSPF interface. Use the **ospf transit-delay** command to set the transit delay for an interface. Enter a number from 0 to 3600 seconds (1 hour). The default is 1 second.

To set the transit delay to 3 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf transit-delay 3
```

To reset the transit delay to the default value of 1 second, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf transit-delay
```

Showing OSPF Information

Use the **show ospf** command to view OSPF information on the CSS. This command is available in all modes. This section includes the following topics:

Showing OSPF Area Information

To show information about OSPF areas, enter:

```
# show ospf areas
```

Table 3-3 describes the fields in the **show ospf areas** command output.

Table 3-3 Field Descriptions for the show ospf areas Command

Field	Description
Area ID	The ID for the area
Type	The area type: Transit or Stub
SPF Runs	The number of times the area calculated the SPF
Area Border Routers	The number of ABRs, including the CSS
AS Boundary Routers	The number of ASBRs, including the CSS, if applicable

Table 3-3 Field Descriptions for the show ospf areas Command (continued)

Field	Description
LSAs	The number of link-state advertisements in the database
Summaries	The capability of summarized LSAs in the stub area, if applicable

Showing Global Statistics

To show OSPF global statistics, enter:

```
# show ospf global
```

Table 3-4 describes the fields in the **show ospf global** command output.

Table 3-4 Field Descriptions for the show ospf global Command

Field	Description
Router ID	The router ID of the CSS.
Admin Status	The state of OSPF on the CSS: Enabled or Disabled.
Area Border Router	Indicates whether the CSS is an ABR. True indicates the CSS is an ABR; otherwise, the field displays False.
AS Boundary Router	Indicates whether the CSS is an ASBR. True indicates the CSS is an ASBR; otherwise, the field displays False.
External LSAs	The number of external LSAs currently contained in the database.
LSA Sent	The number of LSAs sent by the CSS.
LSA Received	The number of LSAs received by the CSS.

Showing IP Interface Information

To show OSPF interfaces, enter:

```
# show ospf interfaces
```

Table 3-5 describes the fields in the **show ospf interfaces** command output.

Table 3-5 Field Descriptions for show ospf interfaces Command

Field	Description
IP Address	The IP address for the OSPF IP interface
Admin State	Administrative state of OSPF on the interface, as affected by the IP interface **ospf enable** command
Area	The area assigned to the interface
Type	The OSPF interface type; always broadcast

Table 3-5 Field Descriptions for show ospf interfaces Command (continued)

Field	Description
State	The functional level of an interface. The state determines whether full adjacencies are allowed to form over the interface. The states include: • **Down** - The initial interface state. In this state, the lower-level protocols indicate the interface is unusable. No protocol traffic is sent or received on the interface. • **Waiting** - The router is trying to determine the identity of the (backup) designated router for the network. To determine the router identify, the router monitors the hello packets it receives. The router is not allowed to elect a backup designated router nor a designated router until it transitions out of the Waiting state. • **DR Other** - The interface is on a network on which another router has been selected to be the designated router. In this state, the router itself has not been selected as the backup designated router. The router forms adjacencies to both the designated router and the backup designated router. • **Backup** - The router itself is the backup designated router on the attached network. The router is the designated router when the present designated router fails. The router establishes adjacencies to all other routers attached to the network. The backup designated router performs slightly different functions during the flooding procedure, as compared to the designated router. • **DR** - The router itself is the designated router on the attached network. Adjacencies are established to all other routers attached to the network. The router must also originate a network LSA for the network node. The network LSA contains links to all routers, including the designated router itself, attached to the network.

Table 3-5 Field Descriptions for show ospf interfaces Command (continued)

Field	Description
Priority	The priority assigned to the interface advertised in the hello packets. When two routers attached to a network both attempt to become the designated router, the router with the highest priority takes precedence. A router whose priority is set to 0 is ineligible to become the designated router on the attached network.
DR	The IP interface address of the designated router selected for the attached network. The designated router is selected on broadcast networks by the hello protocol. Two pieces of identification are kept for the designated router: the Router ID and the IP interface address on the network. The designated router advertises the link state for the network. This network LSA is labeled with the designated router's IP address. The designated router is initialized to 0.0.0.0, which indicates the lack of a designated router.
BR	The backup designated router selected for the attached network. The backup designated router is also selected on all broadcast networks by the hello protocol. All routers on the attached network become adjacent to both the designated router and the backup designated router. The backup designated router becomes the designated router when the current designated router fails. The backup designated router is initialized to 0.0.0.0, indicating the lack of a backup designated router.
Hello	The length of time, in seconds, between the hello packets that the router sends on the interface. This interval is advertised in hello packets sent out on this interface.
Dead	The number of seconds before the router's neighbors declare that the router is down, and when they stop receiving the router's hello packets. This interval is advertised in hello packets sent out on this interface.

Table 3-5 Field Descriptions for show ospf interfaces Command (continued)

Field	Description
Transit Delay	The number of seconds to transmit a Link State Update packet over an interface. LSAs contained in the Link State Update packet have their age incremented by this amount before transmission. This value should take into account transmission and propagation delays; the value must be greater than zero.
Retransmit	The number of seconds between LSA retransmissions for adjacencies belonging to an interface. Also, the interval is used when retransmitting Database Description and Link State Request packets.
Cost	The cost of sending a data packet on the interface, expressed in the link-state metric. The cost of sending a packet is advertised as the link cost for the interface in the router LSA. The cost of an interface must be greater than zero.

Showing Link-State Databases

You can show the entire OSPF link-state database (LSDB) or its specific entry types with the **show ospf lsdb** command. The options for the **show ospf lsdb** command are as follows:

- **show ospf lsdb router** - Displays router LSAs that describe the states of the router interfaces
- **show ospf lsdb network** - Displays network LSAs that describe the set of routers attached to the network
- **show ospf lsdb external** - Displays AS-external LSAs that describe routes to destinations external to the AS
- **show ospf lsdb summary** - Displays summary LSAs that describe summarized routes to the network
- **show ospf lsdb asbr_summ** - Displays summary LSAs that describe routes to AS boundary routers

To show the entire database, enter:

```
# show ospf lsdb
```

Table 3-6 describes the fields in the **show ospf lsdb** command output.

Table 3-6 Field Descriptions for the show ospf lsdb Command

Field	Description
Area	The ID for the area.
Type	The link-state type. The types are as follows: • ASB-Summary for summary LSAs originated by ABRs. The LSAs describe routes to ASBRs. • ASE for AS-external LSAs that describe routes to destinations external to the autonomous system. • Network for the network LSAs that describe the set of routers attached to the network. • Router for router LSAs that describe the collected states of the router interfaces. • Summary-Net for summary LSAs originated by ABRs. The LSAs describe routes to networks.
Link State ID	This field identifies the piece of the routing domain that is being described by the LSA. Depending on the link-state type, the Link State ID has following values: • For the ASB-Summary type, the ID is the router ID of the ASBR. • For the ASE type, the ID is the destination network IP address. • For Network type, the ID is the IP interface address of the network designated router. • For Router type, the ID is the originating router's Router ID. • For Summary-Net type, the ID is the destination network IP address.

Table 3-6 Field Descriptions for the show ospf lsdb Command (continued)

Field	Description
ADV Router	This field specifies the OSPF Router ID of the LSA originator, as follows: • ASB-Summary LSAs, the originators are the ABRs • AS-external LSAs, the originators are ASBRs • Network LSAs, the originators are network-designated routers • Router LSAs, this field is identical to the Link State ID field • Summary LSAs, the originators are the ABRs
Age	The age of the LSA, in seconds. The age is set to 0 when the LSA is originated.
Sequence	A signed 32-bit integer to detect old and duplicate LSAs. The space of sequence numbers is linearly ordered. The larger the sequence number (when compared as signed 32-bit integers), the more recent the LSA. The sequence number 0x80000000 is reserved and unused.
Checksum	The checksum of the complete contents of the LSA, excluding the age field. The age field is excluded to allow the LSA age to increment without updating the checksum. The checksum is used to detect data corruption of an LSA. This corruption can occur while an LSA is being flooded, or while an LSA is being held in a router's memory. The LSA checksum field cannot take on the value of zero; the occurrence of this value is a checksum failure.

Showing ASE Entries

To show AS-external (ASE) entries in the LSDB, enter:

```
# show ospf ase
```

To find specific entries, pipe the output through the **grep** command. For example: **show ospf ase|grep 10.10.10.0**

Table 3-7 describes the fields in the **show ospf ase** command output.

Table 3-7 Field Descriptions for the show ospf ase Command

Field	Description
Link State ID	The network destination for the advertisement
Router ID	The advertising router
Age	The age, in seconds, of the ASE LSA
T	The ASE type of the route; 1 for ASE Type1 or 2 for ASE Type2
Tag	The tag for the route
Metric	The network cost for the route
FwdAddr	The external destination (forwarding address) for the packets

Showing the Configured Advertised ASE Routes

To show the configuration of ASE routes into OSPF, enter:

```
# show ospf advertise
```

To show the configuration of ASE routes into OSPF for a specific host, include the IP address or host and the subnet mask. Enter the address in dotted-decimal format (for example, 192.168.11.1) or mnemonic host-name format (for example, myname.mydomain.com). Enter the mask either:

- As a prefix length in CIDR bit-count notation (for example, /24). Do not enter a space to separate the IP address from the prefix length.
- In dotted-decimal notation (for example, 255.255.255.0).

For example:

```
# show ospf advertise 192.168.11.1/24
```

Table 3-8 describes the fields in the **show ospf advertise** command output.

Table 3-8 Field Descriptions for the show ospf advertise Command

Field	Description
Prefix	The IP address for the route. For the CSS, the prefix is predominately VIP addresses.
Prefix Length	The prefix length for the IP address.
Metric	The network cost for the route. The range is from 1 to 16777215. The default is 1.
Type	The ASE type for the route. By default, the ASE type is ASE type2, which is the external cost to reach the route. ASE type1 combines the external and internal costs.
Tag	The 32-bit tag value to advertise the route. The value is not used by OSPF.

Showing the Redistribution Policy

To show the configured redistribution policy into OSPF, enter:

```
# show ospf redistribute
```

Table 3-9 describes the fields in the **show ospf redistribute** command output.

Table 3-9 Field Descriptions for the show ospf redistribute Command

Static, RIP, Local, or Firewall Field	Description
Routes Redistribution	Indicates whether the redistribution of static, RIP, local or firewall routes is enabled or disabled. If route redistribution is enabled, the configured metric, type, and tag fields are displayed.
Route Metric (displayed when redistribution is enabled)	The external cost for the route. The cost can range from 1 to 16777215. The default is 1.
Route Type (displayed when redistribution is enabled)	The ASE type, either ASE Type1 or ASE Type2. By default, the type is aseType2. The difference between type1 and type2 is how the cost is calculated. For a type 2 ASE, only the external cost (metric) is used when comparing multiple paths to the same destination. For type1 ASE, the combination of the external cost and the cost to reach the ASBR is used.
Route Tag (displayed when redistribution is enabled)	The 32-bit tag value to advertise the external route. The route tag value is not used by the OSPF protocol itself. It is used to communicate information between AS boundary routers.

Showing Summary Route Configuration Information

To show the summary-route configuration information, enter:

```
# show ospf range
```

Table 3-10 describes the fields in the **show ospf range** command output.

Table 3-10 Field Descriptions for the show ospf range Command

Field	Description
Area ID	The ID for the area.
Lsdb Type	The type of link-state database. For an ABR, the type is summaryLink.
Addr Range Mask Range	The address range for the summary route as specified by the IP address (Addr Range) and mask (Mask Range) pair.
Effect	Displays whether the range is advertised or block.

Showing OSPF Neighbors

To show the OSPF neighbors, enter:

```
# show ospf neighbors
```

Table 3-11 describes the fields in the **show ospf neighbors** command output.

Table 3-11 Field Descriptions for show ospf neighbors Command

Field	Description
Address	The IP address of the neighboring router's interface to the attached network. This address is used as the destination IP address when protocol packets are sent as unicasts along this adjacency. The IP address is also used in router LSAs as the Link ID for the attached network if the neighboring router is selected to be the designated router. The CSS learns the neighbor IP address when it receives hello packets from the neighbor.
Neighbor ID	The OSPF Router ID of the neighboring router. The CSS learns the Neighbor ID when it receives hello packets from the neighbor.

Table 3-11 Field Descriptions for show ospf neighbors Command (continued)

Field	Description
Prio	The router priority of the neighboring router. Contained in the neighbor's hello packets, this value is used by OSPF to select the designated router for the attached network.
State/Dr	The state of a conversation being held with a neighboring router. The following states are listed in order of their progression. • Down - The initial state of a neighbor conversation. The Down state indicates that the CSS has received no recent information from the neighbor. • Init - In this state, the CSS has seen a hello packet from the neighbor. However, the CSS has not established bidirectional communication with the neighbor (the router itself did not appear in the neighbor's hello packet). All neighbors in this state (or higher) are listed in the hello packets sent from the associated interface. • 2-Way - In this state, communication between the two routers is bidirectional. The designated router is selected from the set of neighbors in state 2-Way (or greater). • ExStart - This is the first step to create an adjacency between the two neighboring routers. The goal is to decide which router is the master, and to determine the initial Database Description (DD) sequence number. Neighbor conversations in this state (or greater) are called adjacencies.

Table 3-11 Field Descriptions for show ospf neighbors Command (continued)

Field	Description
State/Dr (cont.)	• Exchange - In this state, the CSS sends DD packets to the neighbor to describe its entire link-state database. Each DD packet has a DD sequence number and is explicitly acknowledged. Only one DD packet is allowed to be outstanding at any one time. In this state, the CSS may also send Link State Request packets, requesting the neighbor's more recent LSAs. All adjacencies in Exchange state (or greater) are used by the flooding procedure. In fact, these adjacencies are fully capable of transmitting and receiving all types of OSPF routing protocol packets. • Loading - In this state, the CSS sends Link State Request packets to the neighbor, requesting the more recent LSAs that have been discovered (but not yet received) in the Exchange state. • Full - In this state, the neighboring routers are fully adjacent. These adjacencies will now appear in router LSAs and network LSAs.
Type	Always dynamic.
Rxmt_Q	The number of LSAs to retransmit to the neighbors.

OSPF Configuration in a Startup-Configuration File

The following example shows an OSPF configuration in a startup-configuration file.

```
!************************** GLOBAL **************************
    ospf router-id 121.23.21.1
    ospf enable
    ospf area 1.1.1.1
    ospf as-boundary
    ospf advertise 192.168.4.15 255.255.255.0
    ospf redistribute static
!************************ INTERFACE ************************
interface ethernet-10
    bridge vlan 6
!************************* CIRCUIT *************************
circuit VLAN6
ip address 192.168.2.2 255.255.255.0
    ospf
    ospf area 1.1.1.1
    ospf priority 0
    ospf enable
```

Configuring the Address Resolution Protocol

This chapter describes how to configure Address Resolution Protocol (ARP) to statically configure the IP to Media Access Control (MAC) translations necessary for the CSS to send data to network nodes. You can configure static ARP mapping for any of the CSS Ethernet interface ports.

This chapter contains the following major sections:

ARP Configuration Quick Start

Table 4-1 provides a quick overview of the steps required to configure a static ARP map. 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 4-1.

Table 4-1 ARP Configuration Quick Start

Task and Command Example
1. Define a static ARP mapping. `(config)# arp 192.168.11.1 00-60-97-d5-26-ab e2`
2. Set the time, in seconds, to hold an ARP resolution result. Note that this timeout period affects only dynamic ARP entries. Static ARP entries are permanent and are not affected by this timeout period. `(config)# arp timeout 120`
3. Set the time, in seconds, to wait for an ARP resolution. `(config)# arp wait 15`
4. (Optional) Update the file containing hosts reachable through ARP. `# update arp file` **Note** This command is available only in SuperUser mode.
5. (Optional) Clear ARP parameters for the ARP file or ARP cache that contains known hosts reachable through ARP. `# clear arp file`
6. (Recommended) Display ARP information. For example, to display the complete ARP resolution table, enter: `# show arp`

The following running-configuration example shows the results of entering the commands in Table 4-1.

```
!************************** GLOBAL **************************
  arp 192.168.11.1 00-60-97-d5-26-ab e2
  arp timeout 120
  arp wait 15
```

Configuring ARP

To define a static ARP mapping, use the **arp** command. The syntax for this global configuration mode command is:

arp *ip_or_host mac_address interface* {*vlan*}

The variables and options are as follows:

- *ip_or_host* - The IP address of the system for static mapping. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or in mnemonic host-name format (for example, myhost.mydomain.com).
- *mac_address* - The MAC address of the system mapped to the IP address. Enter the MAC address in hyphenated-hexadecimal notation (for example, 00-60-97-d5-26-ab).
- *interface* - The CSS Ethernet interface port that you want to configure. For a CSS 11501, enter the interface name in *interface port* format (for example, e2). For a CSS 11503 or CSS 11506, the interface format is *slot*/*port* (for example, 3/1).
- *vlan* - The number of the VLAN configured in a trunked interface on which the ARP address is configured (assuming trunking is enabled for the CSS Gigabit Interface port). Enter an integer from 1 to 4094 as the VLAN number.

For example:

```
(config)# arp 192.168.11.1 00-60-97-d5-26-ab e2
```

To remove a static mapping address, use the **no arp** command. For example:

```
(config)# no arp 192.168.11.1
```

Immediately Refreshing the Bridge Forwarding Table for a MAC Down Event

By default, when the CSS receives a Down event for a MAC address in the bridge forwarding table, it may not send an ARP request to an IP address associated with that MAC address for up to 60 seconds to refresh the table. During this time, the bridge flows through the CSS to the MAC address could fail.

You can configure the CSS to immediately send an ARP request for an IP address associated with that MAC address, thus immediately repopulating the entries in bridge forwarding table. Use the global configuration mode **arp mac-down-immediate** command.

For example, enter:

```
(config)# arp mac-down-immediate
```

Note Under certain network conditions, such as STP convergences, the CSS may appear to be causing an ARP storm.

To reset the default behavior, enter:

```
(config)# no arp mac-down-immediate
```

Configuring ARP Timeout

To set the time, in seconds, to hold an ARP resolution result, use the **arp timeout** command. When you change the timeout value, this value affects only new ARP entries. All previous ARP entries retain the old timeout value. This timeout value affects only dynamic ARP entries. Static ARP entries are permanent and are not affected by this timeout.

The timeout value is the number of seconds the CSS holds an ARP resolution result. To set a timeout value, enter an integer from 60 to 86400 (24 hours) seconds. The default is 14400 seconds (4 hours). If you do not want the ARP entries to time out, enter **none** or **86401**.

For example:

```
(config)# arp timeout 120
```

To restore the default timeout value of 14400 seconds, enter:

```
(config)# no arp timeout
```

To remove all entries with the old timeout value, enter the **clear arp cache** command.

Configuring ARP Wait

To set the time, in seconds, to wait for an ARP resolution, use the **arp wait** command. The wait time is the number of seconds the CSS waits for an ARP resolution in response to an ARP request to the network. Enter an integer from 5 to 30 seconds. The default is 5.

For example:

```
(config)# arp wait 15
```

To restore the default wait time of 5 seconds, enter:

```
(config)# no arp wait
```

Updating ARP Parameters

To update the file containing hosts reachable through ARP, use the **update arp** command. This command is available only in SuperUser mode.

For example:

```
# update arp file
```

Clearing ARP Parameters

The CSS enables you to clear ARP parameters for the ARP file or ARP cache. To clear the file that contains known hosts reachable through ARP, use the **clear arp file** command. This command is available only in SuperUser mode.

For example:

```
# clear arp file
```

Use the **clear arp cache** command to delete dynamic entries from the ARP cache. To specify an address for the single ARP entry you want to remove from the ARP cache, use the **clear arp cache** *ip_or_host* command. Enter the address in either dotted-decimal IP notation (for example, 192.168.11.1) or mnemonic host-name format (for example, myhost.mydomain.com).

For example:

```
# clear arp cache 192.168.11.1
```

Showing ARP Information

Use the **show arp** command to display ARP information. To show static ARP mapping when you use the **show arp** command, the IP route must exist in the routing table.

The syntax for this global configuration mode command is:

show arp {**config**|**file**|**management-port**|**summary**|*ip_or_host*}

The syntax and options for the command are as follows:

- **show arp** - Displays the complete ARP resolution table with IP addresses, MAC addresses, and resolution type, excluding entries from the CSS Ethernet management port.
- **config** - Displays ARP global configuration parameters. The screen displays the response timeout and the flush timeout, in seconds.
- **file** - Displays the hosts that are reachable using ARP. The screen displays the IP addresses of the host systems.
- **management-port** - Displays the ARP entries from the CSS Ethernet management port. The ARP resolution table displayed through the **show arp** command displays these entries.

Note The CSS Ethernet management port IP address appears as an entry in the Management Port ARP cache. This is normal CSS behavior.

- **summary** - Displays the total number of static entries, total number of dynamic entries, and total number of entries in the ARP resolution table, excluding the entries from the CSS management port.
- *ip_or host* - The IP address for the system to display its resolution. Enter the address in dotted-decimal format (for example, 192.168.11.1) or mnemonic host-name format (for example, myname.mydomain.com). You cannot enter an ARP entry derived from the CSS Ethernet management port.

For example, to display the complete ARP resolution table, enter:

```
# show arp
```

Table 4-2 describes the fields in the **show arp** command output.

Table 4-2 Field Descriptions for the show arp Command

Field	Description
IP Address	The IP address of the system for ARP mapping.
MAC Address	The MAC address of the system mapped to the IP address.
Type	The resolution type for the entry: Dynamic or Static. The Dynamic resolution type indicates that the entry was discovered through the ARP protocol. The Static resolution type indicates that the entry is from a static configuration.
Port	The CSS interface configured as the egress logical port.

To display a summary of entries in the ARP resolution table, enter:

```
# show arp summary
```

Table 4-3 describes the fields in the **show arp summary** command output.

Table 4-3 Field Descriptions for the show arp summary Command

Field	Description
Static Entry	The total number of static map entries in the ARP resolution table (from a static configuration).
Dynamic Entry	The total number of dynamic map entries in the ARP resolution table (entries discovered through the ARP protocol).
Total Entry	The total number of static and dynamic entries in the ARP resolution table.

To display the global ARP configuration, enter:

```
# show arp config
```

Table 4-4 describes the fields in the **show arp config** command output.

Table 4-4 Field Descriptions for the show arp config Command

Field	Description
ARP Response Timeout	The time, in seconds, to wait for an ARP resolution response before discarding the packet waiting to be forwarded to an address. The time can be from 5 to 30 seconds. The default is 5 seconds.
ARP Flush Timeout	The time, in seconds, to hold an ARP resolution result in the ARP cache. The timeout period can be from 60 to 86400 seconds (24 hours). The default is 14400 seconds (4 hours). An entry of none or 86401 indicates the ARP entries will not timeout.

To display the host IP addresses entered at initialization or boot time through ARP, enter:

```
# show arp file
```

To display the ARP entries from the CSS management port, enter:

```
# show arp management-port
```

Table 4-5 describes the fields in the **show arp management-port** command output.

Table 4-5 Field Descriptions for the show arp management-port Command

Field	Description
IP Address	The IP address of the system for ARP mapping.
MAC Address	The MAC address of the system mapped to the IP address.
Port	The CSS Ethernet management port.

To display the resolution for a host IP address, enter:

```
# show arp 192.50.1.6
```

To display the host IP addresses entered at initialization or boot time through ARP, enter:

```
# show arp file
```

Configuring Routing Information Protocol

The CSS enables you to configure global Routing Information Protocol (RIP) attributes used to advertise routes on the CSS. By default, RIP advertises RIP routes and local routes for interfaces running RIP. The **rip** command advertises other routes.

The timers used by RIP in the CSS include the following default values. These RIP timer values are not user-configurable in the CSS.

- Transmit (Tx) time that is a random value between 15 and 45 seconds (it avoids router synchronization problems
- Route expiration time of 180 seconds (if the CSS loses the link to the next hop router, the route is immediately removed).
- Hold-down time (the amount of time the CSS transmits with an infinite metric) of 120 seconds.

This chapter contains the following major sections:

Note If you prefer OSPF instead of RIP on the CSS, refer to Chapter 3, Configuring Open Shortest Path First, for information on configuring OSPF.

RIP Configuration Quick Start

Table 5-1 provides a quick overview of the steps required to configure global RIP attributes for the CSS. 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 5-1.

Table 5-1 RIP Configuration Quick Start

Task and Command Example
1. Configure the CSS to advertise a route through RIP. `(config)# rip advertise 192.168.1.0/24 9`
2. Configure the CSS to advertise routes from other protocols through RIP (such as firewall routes, OSPF routes, and static routes configured for the Ethernet interface ports). `(config)# rip redistribute static 3`
3. Set the maximum number of routes that RIP can insert into the routing table. `(config)# rip equal-cost 4`
4. (Recommended) Display a RIP configuration for one IP address or all IP addresses configured in the CSS. `(config)# show rip`

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

```
!************************** GLOBAL ***************************
  rip advertise 192.168.1.0 255.255.255.0 9
  rip redistribute static 3
  rip equal-cost 4
```

Configuring RIP Advertise

To advertise a route through RIP on the CSS, use the **rip advertise** command. The syntax for this command is:

rip advertise *ip_address subnet_mask* {*metric*}

The variables for this command are as follows:

- *ip_address* - The IP address for the route prefix. Enter an IP address in dotted-decimal notation (for example, 192.168.1.0).
- *subnet_mask* - The IP prefix length in CIDR bitcount notation (for example, /24) or in dotted-decimal notation (for example, 255.255.255.0).
- *metric* - (Optional) Metric to use when advertising this route. Enter a number from 1 to 15. The default is 1.

For example:

```
(config)# rip advertise 192.168.1.0/24 9
```

Note The network does not have to be present in the routing table to be advertised. The **SNTP ip advertise** command is intended for advertising VIP addresses.

To stop advertising a route through RIP on the CSS, enter:

```
(config)# no rip advertise 192.168.1.0/24
```

Configuring RIP Redistribute

By default, RIP advertises RIP routes and local routes for interfaces running RIP. Use the **rip redistribute** command to advertise routes from other protocols through RIP. This command instructs RIP to advertise other routes, such as firewall routes, OSPF routes, and so on.

The syntax for this command is

rip redistribute [**firewall**|**local**|**ospf**|**static**] {*metric*}

The options and variables for this command are as follows:

- **firewall** - Advertises firewall routes through RIP.
- **local** - Advertises local routes (interfaces *not* running RIP).
- **static** - Advertises static routes configured for the Ethernet interface ports.
- **ospf** - Advertises OSPF routes through RIP.
- *metric* - (Optional) Metric to use when advertising this route. Enter a number from 1 to 15. The default is 1.

For example:

```
(config)# rip redistribute static 3
```

To stop advertising routes from other protocols through RIP, use either the **local, static**, or **firewall** option.

The following commands stop advertising static routes:

```
(config)# no rip redistribute firewall
(config)# no rip redistribute local
(config)# no rip redistribute static
(config)# no rip redistribute ospf
```

Configuring Equal-Cost RIP Routes

To set the maximum number of routes that RIP can insert into the routing table., use the **rip equal-cost** command Enter a number from 1 to 15. The default is 1. For example:

```
(config)# rip equal-cost 4
```

To reset the number of routes to the default value of 1, enter:

```
(config)# no rip equal-cost
```

Showing RIP Configurations

Use the **show rip** command to show a RIP configuration for one IP address or all IP addresses configured in the CSS. This command provides the following options and variables:

- **show rip** - Displays RIP configurations for all interfaces
- **show rip** *ip_address* - Displays a single RIP interface entry
- **show rip globals** - Displays RIP global statistics
- **show rip statistics** - Displays RIP interface statistics for all interfaces
- **show rip statistics** *ip_address* - Displays RIP interface statistics for a specific interface

Table 5-2 describes the fields in the **show rip** command output.

Table 5-2 Field Descriptions for the show rip Command

Field	Description
IP Address	The advertised RIP interface address.
State	The operational state of the RIP interface.
RIP Send	The RIP version that the interface sends. The possible field values are as follows: • **none** - Do not send RIP packets • **RIPv1** - Send RIP version 1 packets only • **RIPv2** - Send RIP version 2 packets only (default)
RIP Recv	The RIP version that the interface receives. The possible values are as follows: • **both** - Receive both version 1 and version 2 (default) • **none** - Receive no RIP packets • **Ripv1** - Receive RIP version 1 packets only • **Ripv2** - Receive RIP version 2 packets only
Default Metric	The default metric used for advertising the RIP interface.

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

Field	Description
Tx Log	The setting for logging RIP packet transmissions (enabled or disabled). The default setting is disabled.
Rx Log	The setting for logging RIP packets received (enabled or disabled). The default setting is disabled.

To display global RIP statistics, enter:

```
# show rip globals
```

Table 5-3 describes the fields in the **show rip globals** command output.

Table 5-3 Field Descriptions for the show rip globals Command

Field	Description
RIP Route Changes	The global number of route changes made to the IP route database by RIP
RIP Query Responses	The global number of query responses sent to RIP query from other systems

To display the RIP interface statistics for all RIP interface entries, enter:

```
# show rip statistics
```

Table 5-4 describes the fields in the **show rip statistics** command output.

Table 5-4 Field Descriptions for the show rip statistics Command

Field	Description
System Route Changes	The global number of route changes made to the IP route database by RIP
System Global Query Responses	The global number of query responses sent to RIP query from other systems
IP Address	The RIP interface IP address
Triggered Updates Sent	The number of triggered RIP updates sent by the interface
Bad Packets Received	The number of bad RIP response packets received by the interface
Bad Routes Received	The number of bad routes in valid RIP packets received by the interface

Configuring the Internet Protocol

This chapter provides information to configure the Internet Protocol (IP) for the CSS and contains the following major sections:

- IP Configuration Quick Start
- Configuring an IP Route
- Disabling an Implicit Service for the Static Route Next Hop
- Configuring an IP Source Route
- Configuring the IP Record Route
- Configuring Box-to-Box Redundancy
- Configuring IP Equal-Cost Multipath
- Forwarding IP Subnet Broadcast Addressed Frames
- Configuring IP Unconditional Bridging
- Configuring IP Opportunistic Layer 3 Forwarding
- Showing IP Configuration Information

For information on configuring static routes for the Ethernet management port, refer to the *Cisco Content Services Switch Administration Guide*.

IP Configuration Quick Start

Table 6-1 provides a quick overview of the steps required to setup the IP configuration for the CSS. 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 6-1.

Table 6-1 IP Configuration Quick Start

Task and Command Example
1. Configure an IP route for the CSS. You can configure a static route, default route, a blackhole route, or a firewall route. For example, to configure a static IP route, enter: `(config)# ip route 192.168.0.0 /16 192.167.1.1`
2. (Optional) If you do not want the CSS to start an implicit service for the next hop of a static route, specify that no implicit service is established to the next hop of the static route. By default the CSS establishes an implicit service for the gateway address when a static route is defined. `(config)# ip no-implicit-service`
3. (Optional) Enable box-to-box redundancy to provide chassis-level redundancy between two identically configured CSSs. `(config)# ip redundancy`
4. (Optional) Set the equal-cost multipath (ECMP) selection algorithm and the preferred reverse egress path. `(config)# ip ecmp address`
5. (Optional) Enable the CSS to forward subnet broadcast addressed frames. `(config)# ip subnet-broadcast`
6. (Recommended) Display IP information for the CSS. For example, to display IP routing information, enter: `# show ip routes`

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

```
!************************** GLOBAL ***************************
  ip no-implicit-service
  ip redundancy
  ip subnet-broadcast

  ip route 192.168.0.0/16 192.167.1.1 1
```

Configuring an IP Route

A static route consists of a destination network address and mask, as well as the next hop to reach the destination. You can also specify a default static route (using 0.0.0.0 as the destination network address and a valid next hop address) to direct frames for which no other destination is listed in the routing table. Default static routes are useful for forwarding otherwise unrouteable packets by the CSS.

When you configure a static route, the CSS creates an internal service that periodically polls the configured next hop address with an ICMP echo (or ping) keepalive. The internal service is called an implicit service. If the router fails, the CSS removes any entries from the routing table that point to the failed router and stops sending network traffic to the failed router. When the router recovers, the CSS:

- Becomes aware of the router
- Reenters applicable routes into the routing table

The implicit service does not determine if the default or static route appears in the routing table. This decision is based on the CSS having a viable ARP entry for the next hop router IP address so the CSS can forward traffic to that destination. The CSS uses the ICMP keepalive as a means to ensure the next hop router MAC address is available and current. However, in certain situations, the next hop router may block ICMP message transmitted by the CSS, which results in a failed ICMP keepalive (the ICMP keepalive is in the Down state). As long as the CSS has the ARP entry of the next hop router the static route is still placed in the routing table.

Note The CSS allows you to disable the internal ICMP keepalive through the **ip-no-implicit service** command. In this case, if the MAC address for the next hop is not known to the CSS the address will not appear in the routing table.

Use the **ip route** command to configure an IP route. You can configure a static route, a default static IP route, a blackhole route (where the CSS drops any packets addressed to the route), or a firewall IP route. Each **ip route** command requires one of the following:

- An IP address and a subnet mask prefix; for example, 192.168.1.0 /24
- An IP address and a subnet mask; for example, 192.168.1.0 255.255.255.0

The syntax for this global configuration command is:

ip route *ip_address subnet_mask*[**blackhole**|*ip_address2*{*distance*|**originated-packets**}|**firewall** *index* {*distance*}]

The syntax and options for the command are as follows:

- *ip_address* - The destination network address. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1).
- *subnet_mask* - The IP subnet mask. Enter the mask in either:
  - CIDR bitcount notation (for example, /24).
  - Dotted-decimal notation (for example, 255.255.255.0).
- **blackhole** - Instructs the CSS to drop any packets addressed to the destination.
- *ip_address2* - The next hop address for the route. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1).
- *distance* - (Optional) The administrative distance. Enter an integer from 1 to 254. A smaller number is preferable. The default value is 1.
- **originated-packets** - Specifies that the route is used only by packets created using flows or sessions going to and from the CSS (for example, a Telnet session to the CSS). The route is not used by flows or sessions that go through the CSS (for example, between an attached server and a remote client).

Note A ping response and an SNMP responses do not use the originated-packets route. A ping *request* sent from the CSS uses the originated-packets route. A ping *response* sent from the CSS does not use the originated-packets route.

- **firewall** - Configures a firewall route. The **firewall** option instructs the CSS to use firewall load balancing for this route. You can optionally set the administrative distance.

  Note The CLI prevents you from configuring IP static routes with identical destinations *and* identical administrative costs, for IP static routes that are firewall routes and IP static routes that are not firewall routes.

- *index* - An existing index number for the firewall route. For information on configuring a firewall index, see the **ip firewall** command (refer to the *Cisco Content Services Switch Security Configuration Guide*).

For example, to configure a static IP route to destination network address *192.168.0.0 /16* and a next hop address of *192.167.1.1*, enter:

```
(config)# ip route 192.168.0.0 /16 192.167.1.1
```

For example, to configure a default IP route using a destination address of *0.0.0.0 /0* and a next hop address of *192.167.1.1*, enter:

```
(config)# ip route 0.0.0.0 /0 192.167.1.1
```

For example, to configure a blackhole route, enter:

```
(config)# ip route 192.168.1.0 /24 blackhole
```

For example, to configure a firewall IP route with an index number of *3* and an administrative distance of *2*, enter:

```
(config)# ip route 192.168.1.0 /24 firewall 3 2
```

To remove a static route, enter:

```
(config)# no ip route 0.0.0.0 /0 10.0.1.1
```

To disable the dropping of packets to a blackhole route, enter:

```
(config)# no ip route 192.168.1.0 /24 blackhole
```

To remove a firewall route, enter:

```
(config)# no ip route 192.168.1.0 /24 firewall 3
```

Disabling an Implicit Service for the Static Route Next Hop

By default, the CSS establishes an implicit (or internal) service for the gateway address when a static route is defined. When you do not want the CSS to start an implicit service for the next hop of a static route, use the **ip no-implicit-service** command. The **ip no-implicit-service** command specifies that no implicit service is established to the next hop of the static route, which disables the internal service ICMP keepalive. In this case, if the ARP address for the next hop is not known to the CSS, the address will not appear in the routing table.

The purpose of the implicit service to the next hop of a static route is to monitor the availability of the next hop to forward data traffic. When the **ip no-implicit-service** command is in effect, traffic is forwarded to the next hop even when the next hop is unavailable. Because of the possibility of data being lost if the next hop becomes unavailable, use of the **ip no-implicit-service** command is strongly discouraged.

Note Static routes can sometimes appear in the CSS routing table even when you have an implicit service for the next hop address (the default setting) and the internal keepalive is down. When the CSS detects the ARP mapping for the next hop in the static route, the CSS continues to list that route in the routing table regardless of the state of the ICMP service keepalive (Down or Up).

When you implement the **ip no-implicit-service** global configuration command, this action does not affect previously configured static routes. The **ip no-implicit-service** command affects only those static routes added after you enable the command. We recommend you reboot the CSS after you modify the configuration to ensure all static routes are the same, which is useful for network monitoring and troubleshooting. If you wish to stop the implicit service for a previously configured static route, then you must delete and reconfigure the static route.

For example:

```
(config)# ip no-implicit-service
```

To reset the default setting, enter:

```
(config)# no ip no-implicit-service
```

Configuring an IP Source Route

To enable the CSS to process frames with information that overrides the default routing, use the **ip source-route** command. For example:

```
(config)# ip source-route
```

Caution Enabling the **ip source-route** command may pose a major security risk to your network. The IP source route specifies information that overrides the default routing a packet would normally take. The packet could then bypass a firewall. If this poses a problem, avoid using the **ip source-route** command.

The CSS does not load balance TCP or UDP packets with IP options that are destined to a VIP address. These packet types are dropped and the CSS returns an ICMP destination/port unreachable error. This behavior exists regardless of the state (enabled or disabled) of the **ip source-route** and **ip record-route** commands.

The CSS, however, does respond to ICMP packets that are destined to a VIP address. The CSS also responds to TCP or UDP packets that include IP options that are destined to a local circuit address, or require that a routing decision be made.

To disable the processing of frames with the IP source-route option (the default behavior), enter:

```
(config)# no ip source-route
```

Configuring the IP Record Route

To enable the CSS to process frames with the IP address of each router along a path, use the **ip record-route** command. For example:

```
(config)# ip record-route
```

Caution Enabling the **ip record-route** command could pose security risks to your network. The **ip record-route** command inserts the IP address of each router along a path into the IP header.

The CSS does not load balance TCP or UDP packets with IP options that are destined to a VIP address. These packet types are dropped and the CSS returns an ICMP destination/port unreachable error. This behavior exists regardless of the state (enabled or disabled) of the **ip record-route** and **ip source-route** commands.

The CSS, however, does respond to ICMP packets that are destined to a VIP address. The CSS also responds to TCP or UDP packets that include IP options that are destined to a local circuit address, or require that a routing decision be made.

To disable the processing of frames with the record-route option (the default behavior), enter:

```
(config)# no ip record-route
```

Configuring Box-to-Box Redundancy

Box-to-box redundancy provides chassis-level redundancy between two identically configured CSSs. Refer to the *Cisco Content Services Switch Redundancy Guide* for information about configuring box-to-box redundancy. Use the **ip redundancy** command to enable box-to-box redundancy.

The CSS does not support simultaneous box-to-box redundancy and VIP or interface redundancy configurations.

For example:

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

To disable box-to-box redundancy, enter:

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

Configuring IP Equal-Cost Multipath

To set the equal-cost multipath (ECMP) selection algorithm and the preferred reverse egress path, use the **ip ecmp** command. The CSS supports a maximum of 15 ECMP paths.

The syntax for this global configuration command is:

ip ecmp [**address**|**no-prefer-ingress**|**roundrobin**]

The options for this global configuration mode command are as follows:

- **address** - Choose among alternate paths based on IP addresses. For example:

  ```
  (config)# ip ecmp address
  ```

- **no-prefer-ingress** - Do not prefer the ingress path of a flow for its reverse egress path. By default, the ingress path for a flow is the preferred egress path. This means that the preferred interface over which to reply to a client is the interface on which the CSS originally received the request from the client. Note that this command option has no effect on UDP traffic.

  For example:

  ```
  (config)# ip ecmp no-prefer-ingress
  ```

  To reset the ingress path of a flow for its preferred reverse egress path, enter:

  ```
  (config)# no ip ecmp no-prefer-ingress
  ```

- **roundrobin** - Alternate between equal paths in roundrobin fashion. For example:

  ```
  (config)# ip ecmp roundrobin
  ```

Note The CSS applies the ECMP selection algorithm for non-TCP/UDP packets (for example, ICMP) on a packet-by-packet basis. Multipath selection for TCP and UDP is performed on a per-flow basis, and all packets for a particular flow take the same path.

Forwarding IP Subnet Broadcast Addressed Frames

To enable the CSS to forward subnet broadcast addressed frames, use the **ip subnet-broadcast** command.

For example:

```
(config)# ip subnet-broadcast
```

To disable forwarding of subnet broadcast addressed frames (the default behavior), enter:

```
(config)# no ip subnet-broadcast
```

Caution Enabling the CSS to forward the subnet broadcast can make the subnet susceptible to "smurf" attacks; an attacker sends an ICMP echo request frame using a subnet broadcast address as a destination and a forged address as the source.

If a "smurf" attack is successful, all the destination subnet hosts reply to the echo and flood the path back to the source. By disabling subnet broadcast forwarding, the original echo never reaches the hosts.

Configuring IP Unconditional Bridging

By default, the routing table lookup of a destination path by the CSS on received packets overrides bridging decisions to be made for those packets. If the routing table specifies that the CSS use a different physical Ethernet port than what is specified for port bridging, the CSS ignores the bridging decision. If you have a network that you want to bridge through the CSS to an upstream router, you may want to force the CSS to make a bridging decision on the received packets instead of making a routing table decision.

Use the **ip uncond-bridging** global configuration command to always make a bridging decision on the received packets. With this command, the bridging decision always takes precedence over a routing table decision.

For example:

```
(config)# ip uncond-bridging
```

To restore the default behavior of the CSS, enter:

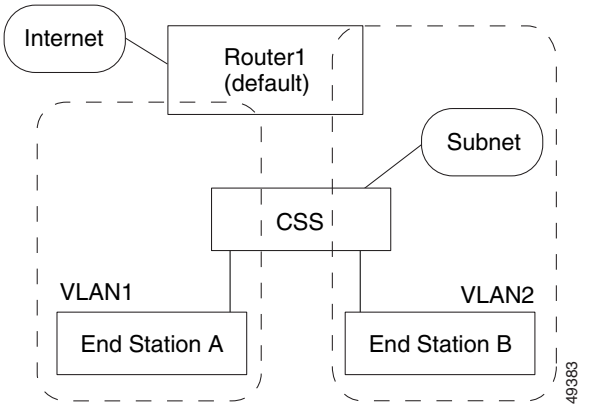
```
(config)# no ip uncond-bridging
```

Configuring IP Opportunistic Layer 3 Forwarding

The CSS opportunistic Layer 3 forwarding feature allows the CSS to reduce the number of network device hops for certain packets or flows. The CSS forwards packets at Layer 3 if the destination MAC address in the Ethernet header is the CSS MAC address. Use the **ip opportunistic** command to enable opportunistic Layer 3 forwarding and allow the CSS to make Layer 3 forwarding decisions even if the Layer 2 packet destination MAC address does not belong to the CSS.

For example, Figure 6-1 shows a CSS connected to VLAN1 and VLAN2. Each VLAN has an end station and an uplink to Router1. End stations A and B both point to Router1 as their default router. When End Station A transmits a packet to End Station B, it uses its default route to Router1. The packet contains Router1's destination MAC address. A traditional Layer 2 device forwards the packet to Router1, and Router1 forwards the packet to End Station B on VLAN2.

Figure 6-1 Example of Opportunistic Layer 3 Forwarding



Using opportunistic Layer 3 forwarding, the CSS inspects the IP packet header to determine the destination IP address. Instead of forwarding the packet to Router1, the CSS forwards the packet directly to End Station B. Because the CSS handles the packet only once, the router and uplink are not used and network resources are conserved.

The options for this global configuration mode command are as follows:

- **local (default)** - Applies opportunistic Layer 3 forwarding if the destination IP address belongs to a node that resides on one of the subnets directly attached to the CSS *and* the CSS is aware of an ARP resolution for that node. Because the local option is the default, use the **no ip opportunistic** command to reconfigure IP opportunistic Layer 3 forwarding to the local setting.
- **all** - Applies opportunistic Layer 3 forwarding if the destination IP address matches any entry in the CSS routing table. We do not recommend this option if the topology includes multiple routers and the CSS does not know all of the routes the routers are aware of.
- **disabled** - The CSS does not perform opportunistic Layer 3 forwarding. Regular Layer 3 forwarding is performed only for packets that contain the CSS destination MAC address.

For example, to configure IP opportunistic Layer 3 forwarding to **all**, enter:

```
(config)# ip opportunistic all
```

To reconfigure IP opportunistic Layer 3 forwarding to the default of **local** enter:

```
(config)# no ip opportunistic
```

When you configure **ip opportunistic all**, you can use the **ip route originated-packets** command (see the "IP Configuration Quick Start" section) to configure routes that the CSS uses to reach devices, but does not use as opportunistic routes for forwarding traffic. Routes created using the **ip route originated-packets** command apply only to packets that originate on the CSS. Packets and flows forwarded by the CSS do not use these routes.

For example:

```
(config)# ip route 0.0.0.0 /0 192.168.1.7 originated-packets
```

Configuring Advanced Route Remapping

To configure a CSS to remap flows using the best available route, use the **ip advanced-route-remap** command. The syntax of this global configuration mode command is:

ip advanced-route-remap

For example, enter:

```
(config)# ip advanced-route-remap
```

To disabled the remapping of flows using the best available route, enter:

```
(config)# no ip advanced-route-remap
```

Showing IP Configuration Information

Use the **show ip** command to display IP information for the CSS. This section includes the following topics:

- Showing IP Global Configuration Parameters
- Showing IP Interface Information
- Showing IP Routing Information
- Showing IP Statistics
- Showing a Summary of IP Global Statistics

Showing IP Global Configuration Parameters

Use the **show ip config** command to display IP global configuration parameters. These parameters show the state (enabled or disabled) of the source route option, forward IP broadcasts, record-route option, and IP route change logging. The **show ip config** command also shows the value for the orphaned route timer.

Table 6-2 describes the fields Zin the **show ip config** output.

Table 6-2 Field Descriptions for the show ip config Command

Field	Description
Source Route Option	Indicates whether processing of source-routed frames is enabled or disabled.
Forward IP Broadcasts	Indicates whether forwarding IP broadcasts is enabled or disabled.
Orphaned Route Timer	The setting for the orphaned route timer.
Record Route Option	Indicates whether processing with the record-route option is enabled or disabled.
Multiple Equal Cost Path Algorithm	The setting for the equal-cost multipath selection algorithm. The possible settings are as follows: • **Address** - Choose among alternate paths based on IP addresses • **Roundrobin** - Alternate between equal paths in roundrobin fashion
IP Route Change Logging	Indicates whether logging IP route changes is enabled or disabled.

Showing IP Interface Information

Use the **show ip interfaces** command to display configured IP interfaces on the CSS. The display includes the circuit state, IP address, broadcast address, Internet Control Message Protocol (ICMP) settings, and Router Discovery Program (RDP) settings.

Table 6-3 describes the fields in the **show ip interfaces** command output.

Table 6-3 Field Descriptions for the show ip interfaces Command

Field	Description
Circuit Name	The name of the circuit associated with the IP interface.
State	The state of the IP interface. The possible states are as follows: • **Active (1)** - Interface is up • **Disabled (2)** - Interface is disabled • **NoCircuit (3)** - Interface is waiting for an underlying circuit
IP Address	The IP address assigned to the circuit.
Network Mask	The network mask of the circuit.
Broadcast Address	The broadcast IP address associated with the IP interface. If left at zero, the all-ones host is used for numbered interfaces. 255.255.255.255 is always used for unnumbered interfaces.
Redundancy	Indicates whether the redundancy protocol is running on the interface. The default state is Disabled.
ICMP Redirect	Whether the transmission of Internet Control Message Protocol (ICMP) redirect messages is enabled or disabled. The default state is Enabled.
ICMP Unreachable	Whether the transmission of ICMP Destination Unreachable messages is enabled or disabled. The default state is enabled.
RIP	Whether RIP is enabled or disabled.

Showing IP Routing Information

Use the **show ip routes** command to display IP routing information. The syntax and options for this command are as follows:

- **show ip routes** - Displays the entire routing table, including host IP address, next hop, interface, route type, protocol, age (in seconds), and metric.
- **show ip routes firewall** - Displays all firewall routes.
- **show ip routes local** - Displays all local routes.
- **show ip routes ospf** - Displays all OSPF routes.
- **show ip routes rip** - Displays all RIP routes.
- **show ip routes static** - Displays all static routes.
- **show ip routes summary** - Displays the total number of OSPF routes (including a breakdown of Intra, Inter, and Ext routes), RIP routes, local routes, static routes, and firewall routes.
- **show ip routes** *ip_or_host* {**to** *ip_or_host* | *mask_or_prefix*} - Displays information about a route to a destination, a specific route, or routes in a range.

The variables are as follows:

- *ip_or_host* - The IP address of the host or network prefix. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1). The IP address after the **to** keyword is the final IP address in a range.
- *mask_or_prefix* - Subnet address of the specific network. Enter the subnet address in mask or prefix notation (for example, /24).

To show all IP routes in the CSS, enter:

```
# show ip routes
```

Table 6-4 describes the fields in the **show ip routes** command output.

Table 6-4 Field Descriptions for the show ip routes Command

Field	Description
Prefix/length	The IP address and prefix length for the route.
Next hop	The IP address for the next hop.

Table 6-4 Field Descriptions for the show ip routes Command (continued)

Field	Description
If	The Index value that identifies the local interface through which the next hop of this route should be reached.
Type	The type of the route entry. The possible types are as follows: • local - Local interface • remote - Remote destination • mgmt - Management interface
Proto	The protocol for the route.
Age	The maximum age of the route.
Metric	The metric cost of the route.

Showing IP Statistics

Use the **show ip statistics** command to display aggregate TCP statistics for the CSS or module in a CSS 11503 or 11506 chassis. The syntax for this command is:

show ip statistics {*slot_number*}

The optional *slot_number* variable is the slot number for the module in the CSS. This variable allows you to display the statistics only for the module in the specified slot. If you do not specify a slot number, this command displays the statistics for all modules in the chassis.

Table 6-5 describes the fields in the **show ip statistics** output.

Table 6-5 Field Descriptions for the show ip statistics Command

Field	Description
UDP Statistics	
Input Datagrams	The total number of flow-related UDP datagrams delivered to UDP users.
No Port Errors	The total number of received UDP datagrams for which there was no application at the destination port.

Table 6-5 Field Descriptions for the show ip statistics Command (continued)

Field	Description
Output Datagrams	The total number of flow-related UDP datagrams sent from the CSS.
Input Errors	The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.
TCP Statistics	
Retransmit Algorithm	The algorithm used to determine the timeout value for retransmitting unacknowledged octets.
Max Retransmit Time	The maximum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds.
Active Opens	The number of times TCP connections have made a direct transition to the SYN-SENT state from the Closed state.
Failed Attempts	The number of times TCP connections have made a direct transition to the Closed state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times TCP connections have made a direct transition to the Listen state from the SYN-RCVD state.
Established Conns	The number of TCP connections for which the current state is either Established or Close-Wait.
Output Segments	The total number of segments sent, including those on current connections but excluding those containing only retransmitted octets.
Input Errors	The total number of segments received in error (for example, bad TCP checksums).
Min Retransmit Time	The minimum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds.
Max TCP Connections	The total number of TCP connections that the CSS supports.

Table 6-5 Field Descriptions for the show ip statistics Command (continued)

Field	Description
Passive Opens	The number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.
Resets	The number of times TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.
Input Segments	The total number of segments received, including those received in error. This count includes segments received on currently established connections.
Retransmit Segments	The total number of segments retransmitted; that is, the number of TCP segments transmitted containing one or more previously transmitted octets.
Output Resets	The number of TCP segments sent containing the RST flag.
ICMP Statistics	
Echo Requests In	The number of received ICMP Echo request messages. Typically, when the CSS receives the ICMP request, both the Echo Requests In and the Echo Replies Out counters increment as a pair for the ICMP request in and ICMP reply out packets.
Echo Replies In	The number of received ICMP Echo reply messages. Typically, when the CSS receives an ICMP reply, both the Echo Requests Out and the Echo Replies In counters increment as a pair for the ICMP reply in and ICMP request out packets.
Unreachable	The number of received ICMP Destination Unreachable messages.
Redirect	The number of received ICMP Redirect messages.
Router Solicit	The number of received ICMP router solicitation packets.
Param Problem	The number of received ICMP Parameter Problem messages.

Table 6-5 Field Descriptions for the show ip statistics Command (continued)

Field	Description
Timestamp Reply	The number of sent ICMP Timestamp Reply messages.
Information Reply	The number of received ICMP information reply packets.
Mask Reply	The number of received ICMP Address Mask Reply messages.
Echo Requests Out	The number of transmitted ICMP Echo request messages. Typically, when the CSS transmits an ICMP request, both the Echo Requests Out and the Echo Replies In counters increment as a pair for the ICMP request out and ICMP reply in packets.
Echo Replies Out	The number of transmitted ICMP Echo reply messages.Typically, when the CSS transmits an ICMP reply, both the Echo Requests In and the Echo Replies Out counters increment as a pair for the ICMP reply out and ICMP request in packets.
Source Quench	The number of received ICMP Source Quench messages.
Router Adv	The number of received ICMP router advertisement packets.
Time Exceeded	The number of received ICMP Time Exceeded messages.
Timestamp	The number of sent ICMP Timestamp (request) messages.
Information Request	The number of received ICMP information request packets.
Mask Request	The number of sent ICMP Address Mask Request messages.
Invalid	The number of received bad ICMP type packets.
ARP Statistics	
Requests In	The number of received ARP request packets.
Requests Out	The number of sending ARP request packets.

Table 6-5 Field Descriptions for the show ip statistics Command (continued)

Field	Description
Duplicate Addr	The number of received ARP packets with a detected duplicate IP address. The duplicate IP address can be the local IP address, VIP, or virtual interface.
Invalid	The number of invalid or bad ARP packets.
Replies In	The number of received ARP reply packets.
Replies Out	The sending ARP reply packet count.
In Off Subnet	The number of received ARP packets with sender or target addresses outside of the subnet range of the receiving interface.
Unresolved	The number of processed IP frames with unresolved next hop MAC addresses.

Resetting IP Statistics

To set the global IP (TCP/UDP) statistics for the CSS to zero, use the **zero ip statistics** command in any mode. This command sets the TCP/UDP statistics displayed by the **show ip statistics** command to zero. For more information about the show ip statistics command, see the "Showing IP Statistics" section.

Showing a Summary of IP Global Statistics

Use the **show ip summary** command to display a summary of IP global statistics. The statistics include data on reachable and total routes, reachable and total hosts, memory in use for each, and total IP routing memory in use.

Table 6-6 describes the fields in the **show ip summary** command output.

Table 6-6 Field Descriptions for the show ip summary Command

Field	Description
Reachable Routes	The current number of reachable routes.
Total Routes	The current number of routes maintained, both reachable and unreachable.
Reachable Hosts	The current number of reachable host entries.
Total Hosts	The current number of host entries, both reachable and unreachable.
Total Memory in use - IP Routing Memory Pool	The total amount of memory in bytes allocated for the IP routing table. When there are no additional free entries in the memory pool, more memory is allocated to the pool.

Configuring the Cisco Discovery Protocol

The Cisco Discovery Protocol (CDP) is a medium-independent protocol that runs over Layer 2 (the data link layer) on the CSS and other Cisco manufactured equipment, such as routers, switches, bridges, and access servers. Use the **cdp** global configuration command to allow the CSS to advertise itself to all other neighboring Cisco CDP-compatible devices on a network. The CSS only transmits CDP advertisements to other CDP-compatible devices on the network; the CSS does not listen for CDP messages from the other CDP-compatible devices, and does not maintain a CDP table.

Any Cisco device with CDP support can learn about the CSS by listening to the periodic messages transmitted by the CSS and determining when the CSS is active. Network operators and analysts can use this information for configuration monitoring, topology discovery, and fault diagnosis.

CDP messages contain specific information about the CSS, such as:

- Device ID (CSS base MAC address)
- IP address (CSS management port IP address)
- Ethernet port ID name
- CSS functional capability flag (Router, Transparent Bridge, or Switch)
- CSS software version
- CSS platform

CDP advertisements also include hold time information, which defines the length of time the receiving device is to hold CDP information before discarding it.

This chapter contains the following major sections:

CDP Configuration Quick Start

Table 7-1 provides a quick overview of the steps required to configure CDP for the CSS. 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 7-1.

Table 7-1 CDP Configuration Quick Start

Task and Command Example
1. Enable CDP transmissions from the CSS to other neighboring Cisco CDP-compatible devices on the network. `(config)# cdp run`
2. Specify the amount of time a receiving device retains the CDP information sent by the CSS (time-to-live information) before discarding this information. `(config)# cdp holdTime 255`
3. Specify the frequency at which the CSS transmits CDP packets to all receiving CDP-compatible devices. `(config)# cdp timer 120`
4. (Recommended) Display and verify CDP information for the CSS. `(config)# show cdp`

The following running-configuration example shows the results of entering the commands in Table 7-1.

```
!************************** GLOBAL ***************************
  cdp run
  cdp holdTime 255
  cdp timer 120
```

Enabling CDP

By default, CDP is disabled for the CSS. Use the **cdp run** global configuration command to enable CDP transmissions from the CSS to other neighboring Cisco CDP-compatible devices on the network.

For example:

```
(config)# cdp run
```

To disable CDP transmissions on the CSS, enter:

```
(config)# no cdp run
```

Setting the CDP Hold Time

The CDP hold time is the amount of time a receiving device retains the CDP information sent by the CSS (time-to-live information) before discarding this information. If a neighboring device does not receive a CDP message before the hold time expires, the neighboring device drops the CSS as a neighbor. By default, the hold time is 180 seconds. To specify the hold time, use the **cdp holdTime** global configuration command. Valid entries are 10 to 255 seconds.

To specify a CDP hold time of 255 seconds for the receiving device, enter:

```
(config)# cdp holdTime 255
```

To reset the CDP hold time back to the default value of 180 seconds, enter:

```
(config)# no cdp holdTime
```

Setting the CDP Transmission Rate

By default, the frequency at which the CSS transmits CDP packets to all receiving CDP-compatible devices is 60 seconds. To specify the frequency at which the CSS transmits CDP packets to all receiving CDP-compatible devices, use the **cdp timer** global configuration command. Valid entries are 5 to 254 seconds.

To change the CDP transmission rate for the CSS to 120 seconds, enter:

```
(config)# cdp timer 120
```

To reset the CDP timer to the default rate of 60 seconds, enter:

```
(config)# no cdp timer
```

Showing CDP Information

Use the **show cdp** command to display and verify CDP information for the CSS, such as frequency of transmissions and the hold time for transmitted CSS CDP information.

For example:

```
(config)# show cdp

Global CDP information:
Sending CDP packets every  60 seconds
Sending a holdtime value of 16 seconds
TimeLastCdpSent: 0 days 00:00:30
```

The following example illustrates the CDP output on a Cisco Catalyst 8540 router using the Cisco IOS **show cdp neighbors** command.

```
24-8540-1>show cdp neighbors

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge S - Switch, H -
Host, I - IGMP, r - Repeater

Device ID          Local Intrfce  Holdtme Capability Platform  Port ID
00-10-58-01-4d-e3   Eth 0            178   R T S     CSS 11050 Eth-Mgmt
SCA043801A5         Eth 0            144   T S       WS-C6009  3/1
25-8540-1           Fas 0/0/7        142   R T       C8540CSR  Fas 0/0/4
25-8540-1           Eth 0            142   R T       C8540CSR  Eth 0
SCA043801HU(bxb11   Eth 0            151   T S       WS-C6009  2/48
00-07-85-43-14-1d   Eth 0            170   R T S     CSS11503  Eth-Mgmt
```

Configuring the DHCP Relay Agent

The Dynamic Host Configuration Protocol (DHCP) servers provide configuration parameters to DHCP clients. When DHCP clients and associated servers do not reside on the same IP network or subnet, a DHCP relay agent can transfer DHCP messages between them. To configure a DHCP relay agent on a CSS, define DHCP server destinations on a circuit and enable the DHCP relay agent on the circuit.

You must first assign an IP address on the circuit to be able to configure the DHCP relay agent for the circuit. Use the **ip address** command in the specific circuit mode to assign the IP address and a subnet mask. For example:

```
(config-circuit[VLAN2])# ip address 178.3.6.53/8
```

This chapter contains the following major sections:

DHCP Relay Agent Configuration Quick Start

Table 8-1 provides a quick overview of the steps required to configure the DHCP relay agent for the circuit. 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 8-1.

Table 8-1 DHCP Relay Agent Configuration Quick Start

Task and Command Example
1. Specify the DHCP relay destination IP address in dotted-decimal notation. `(config-circuit[VLAN2])# dhcp relay-to 192.168.22.25`
2. Enable the DHCP relay agent on the CSS circuit. `(config-circuit[VLAN2])# dhcp-relay-agent`
3. Set the maximum allowable number in the hops field of the BOOTP header. `(config)# dhcp-agent max-hops 10`
4. (Optional) Verify the DHCP configuration. `(config)# # show dhcp-relay-agent global`

The following running-configuration example shows the results of entering the commands in Table 8-1.

```
!************************** GLOBAL ***************************
    dhcp-agent max-hops 10


!************************** CIRCUIT **************************
circuit VLAN2
      dhcp relay-to 192.168.22.25
      dhcp-relay-agent
```

Adding a DHCP Destination on a Circuit

A CSS circuit acts as the DHCP relay agent. For each circuit on the CSS, you can configure a maximum of five DHCP destinations. The initial DHCP broadcast request is sent to all of the configured destinations.

Do not configure a relay destination on a circuit when the relay destination is directly connected to or reachable from one of the ports on the same circuit. In this case, the DHCP packets reach the relay destination through normal broadcast and a relay agent is not required.

Use the **dhcp relay-to** command to specify the DHCP relay destination address. This command is available in circuit configuration mode. Enter an IP address in dotted-decimal notation.

For example, to add a destination address of 192.168.22.25 to a DHCP server, enter:

```
(config-circuit[VLAN2])# dhcp relay-to 192.168.22.25
```

To remove the relay destination address, enter:

```
(config-circuit[VLAN2])# no dhcp relay-to 192.168.22.25
```

Enabling and Disabling DHCP on the Circuit

After you enable the DHCP relay agent on the CSS circuit, the CSS transfers DHCP messages between DHCP clients and servers. Use the **dhcp-relay-agent** command to enable the agent on the circuit. This command is available in circuit configuration mode.

For example:

```
(config-circuit[VLAN2])# dhcp-relay-agent
```

To disable the DHCP relay agent on the circuit, enter:

```
(config-circuit[VLAN2])# no dhcp-relay-agent
```

Defining the Hops Field Value for Forwarding DHCP Messages

The CSS forwards or discards a DHCP message based on the hops field value in the BOOTP header. When messages have values in the hops fields that exceed the maximum value set on the CSS, the CSS discards the message. Use the **dhcp-agent max-hops** global configuration command to set the maximum allowable number in the hops field. By default, the maximum allowable number is 4. You can set a number from 1 to 15.

For example, to set the maximum allowable value of 10, enter:

```
(config)# dhcp-agent max-hops 10
```

To reset the maximum allowable number in the hops field to the default of 4, enter:

```
(config)# no dhcp-agent max-hops
```

Displaying the DHCP Relay Configuration

Use the **show dhcp-relay-agent global** command to display the DHCP configuration information on the CSS. This command is available in all modes. For example:

```
# show dhcp-relay-agent global
```

Table 8-2 describes the fields in the **show dhcp-relay-agent global** command output.

Table 8-2 Field Descriptions for the show dhcp-relay-agent global Command

Field	Description
Max Hops	The maximum allowable number in the hops field of the BOOTP header. The CSS does not forward packets with headers that contain a larger number.
Number of circuits configured for DHCP	The number of CSS circuits configured for DHCP.

Table 8-2 Field Descriptions for the show dhcp-relay-agent global Command (continued)

Field	Description
Circuit	The circuit configured for DHCP.
IfAddress	The interface address for the circuit.
DHCP State	The DHCP relay agent state on the circuit (Enabled or Disabled).
Relay destination	The DHCP relay destination address for the server. Each circuit can have five destination addresses.

A

B

C

D

E

F

H

I

O

P

Q

R

S

T

V

W

Z