



Configuring Box-to-Box Redundancy

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

This chapter contains the following major sections:

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

Overview of CSS Redundancy

Redundancy helps to ensure:

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

A CSS provides three types of redundancy.

- Virtual IP (VIP) redundancy and virtual interface redundancy - Provide redundant VIP addresses and redundant virtual interfaces for fate sharing and server default gateways. For details, refer to [Chapter 6, Configuring VIP and Virtual Interface Redundancy](#).
- Adaptive Session Redundancy (ASR) - Provides session-level redundancy (stateful failover) to continue active flows without interruption if the master CSS fails over to the backup CSS. For details, refer to [Chapter 7, Configuring Adaptive Session Redundancy](#).
- Box-to-box redundancy - Provides chassis-level redundancy between two identically configured CSSs. For details, see this chapter.

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

When to Use VIP and Virtual Interface Redundancy

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

Configure VIP redundancy:

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

Configure interface redundancy:

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

When to Use ASR

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

Configure ASR:

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

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

When to Use Box-to-Box Redundancy

Configure box-to-box redundancy when you:

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

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

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

Redundancy Protocol Overview

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

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

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

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

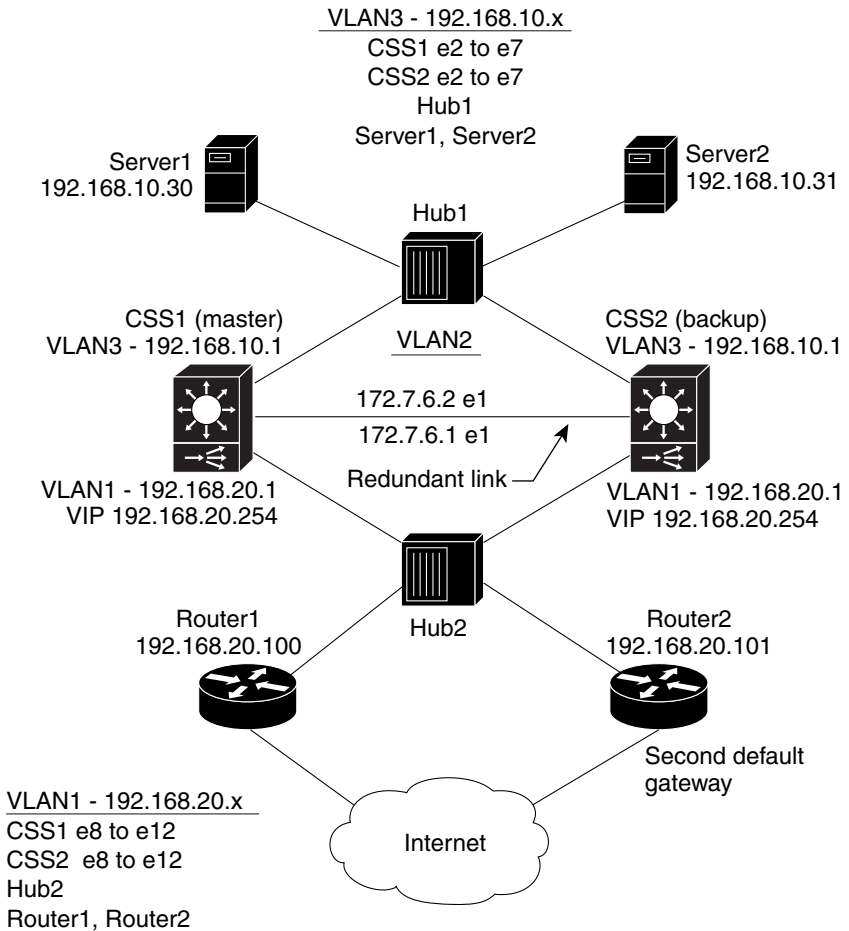
If a former master comes back online, it becomes a backup CSS automatically when it receives the master CSS messages, unless you explicitly designated the CSS to be the master when you configured it. For details on IP redundancy, see [“Configuring IP Redundancy”](#) later in this chapter.

**Caution**

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

Figure 8-1 shows an example of a redundant configuration with multiple VLANs. Table 8-1 uses this figure to define the command examples necessary to configure the redundancy protocol.

Figure 8-1 Redundancy Configuration Example



49641

Redundancy Configuration Quick Start

Table 8-1 provides the steps required to configure the redundant configuration shown in Figure 8-1. Each step includes the CLI command required to complete the task. For a complete description of each feature, refer to the sections following Table 8-1.

Listed below are the basic steps to configure redundancy:

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

**Caution**

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

**Note**

You must connect the crossover cable directly to the Gigabit Ethernet (GE) ports (11500 series CSS with software version 7.10.1.02 and greater) or the Fast Ethernet (FE) ports on the redundant CSSs. Do not use Layer 2 devices between the two CSSs on the redundant link.

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

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

**Note**

If you make configuration changes to the master CSS startup-config, you must make the same changes to the backup CSS startup-config. To learn how to synchronize the running-configs of the master CSS and the backup CSS, see [“Synchronizing a Redundant Configuration”](#) later in this chapter.

Table 8-1 Redundancy Configuration Quick Start

Task and Command Example

1. Install the crossover cable before you power up the CSSs. Make the CSS-to-CSS connection using a Category 5 crossover cable. This table uses port e1 (ethernet-1) on the master CSS and port e1 on the backup CSS.

2. Configure each server's default gateway as the CSS circuit VLAN IP address.

3. Enter the **ip redundancy** command on the master CSS to enable CSS-to-CSS redundancy.

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

4. Configure an interface on the master CSS for a redundant connection to the backup CSS. For information on configuring interfaces, refer to the *Cisco Content Services Switch Administration Guide*.

```
(config)# interface e1
```

5. Assign the interface to the redundant connection VLAN. For information on bridging the interface to a VLAN, refer to the *Cisco Content Services Switch Administration Guide*.

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

6. Enter circuit mode for the redundant VLAN. For information on configuring circuits, refer to the *Cisco Content Services Switch Administration Guide*.

```
(config-if[e1])# circuit VLAN2  
(config-circuit[VLAN2])#
```

7. Assign an IP address and subnet mask to circuit VLAN2. For information on configuring a circuit IP address, refer to the *Cisco Content Services Switch Administration Guide*.

```
(config-circuit[VLAN2])# ip address 172.7.6.1/24
```

8. Enable the redundancy protocol on the redundant IP interface.

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

Table 8-1 Redundancy Configuration Quick Start (continued)**Task and Command Example**

9. Define the other circuits as redundant circuits.

```
(config-if[e1])# circuit VLAN1
(config-circuit[VLAN1])# redundancy
(config-if[e1])# circuit VLAN3
(config-circuit[VLAN3])# redundancy
```

10. From SuperUser mode, save the master CSS running-config to the startup-config.

```
# copy running-config startup-config
```

11. FTP the startup-config to a PC for editing.

12. Using a text editor, edit the startup-config by including the backup CSS circuit VLAN IP address for the redundant connection (in Figure 8-1, circuit VLAN2, IP address 172.7.6.2/24). Do not change the backup CSS VIP. The master and backup CSS must have the same VIP. If you have multiple VIPs, you must configure them on both the master and backup CSSs.

13. FTP the new file to the backup CSS and, if necessary, rename it as startup-config.

14. Reboot the backup CSS.

15. Enter the **show redundancy** command to display the redundancy configuration and ensure that the backup CSS is configured properly.

16. Cable all hubs (or switches) to the backup CSS.

Cabling Redundant CSS Switches

When you set up a redundant configuration, install a Category 5 crossover cable directly to the CSSs to connect the master and backup interfaces.



Caution

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



Note

You must connect the crossover cable directly to the Gigabit Ethernet (GE) ports (11500 series CSS with software version 7.10.1.02 and greater) or the Fast Ethernet (FE) ports before you power on the redundant CSSs. Do not use Layer 2 devices between the two CSSs on the redundant link.

[Table 8-2](#) lists the pinouts for the CSS Fast Ethernet connectors and the crossover pinouts.

Table 8-2 RJ-45 Fast Ethernet Connector Pinouts

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

Configuring Redundancy

Configuring the redundancy protocol requires you to:

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

**Note**

The CSS does not support IP redundancy and VIP redundancy simultaneously. For information on VIP redundancy, refer to [Chapter 6, Configuring VIP and Virtual Interface Redundancy](#).

Configuring IP Redundancy

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

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

**Note**

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

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

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

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

The syntax for this global configuration mode command is:

- **ip redundancy** - Enables CSS-to-CSS redundancy on the backup CSS. If you have no requirement to define a CSS as the master CSS, use this command on both CSSs in the redundant configuration.
- **ip redundancy master** - Enables CSS-to-CSS redundancy on the CSS that you want to designate as the master CSS. (Be sure to issue **ip redundancy** on the backup CSS.) You can issue **ip redundancy master** on a CSS:
 - Whether or not it was initially booted as the master or the backup. If you issue this command on the backup CSS, it becomes the master and the other CSS becomes the backup CSS automatically.
 - When CSS-to-CSS redundancy is currently enabled.

For example:

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

**Caution**

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

To disable CSS-to-CSS redundancy, enter:

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

**Note**

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

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

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

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

To unassign the CSS as the master CSS, enter:

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



Note

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

Configuring Redundant Circuits

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



Note

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

When you configure redundancy, configure it on circuits (VLANs) that contain network IP addresses shared by the redundant CSSs (in [Figure 8-1](#), these are VLAN1 and VLAN3). Do not configure redundancy on the circuit (VLAN) configured specifically for redundancy communications (in [Figure 8-1](#), this is VLAN2).

The example below configures VLAN1 as a redundant circuit, which contains a shared network IP address (in [Figure 8-1](#), this is 192.168.20.1).

For example:

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

To remove a circuit from the redundancy configuration, enter:

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

Configuring the Redundancy Protocol

To configure the redundancy protocol on the circuit (VLAN) connecting the two CSSs, enter the **redundancy-protocol** command in IP interface configuration mode. Enter the command for the circuit you configured specifically for redundancy communications (in [Figure 8-1](#), this is VLAN2).

**Note**

The CSS box-to-box redundancy protocol is supported on CSS 11500 Gigabit Ethernet (GE) ports in software version 7.10.1.02 and greater.

For example:

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

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

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

Configuring the VRRP Backup Timer

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

**Note**

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

The syntax for this global configuration mode command is:

```
vrrp-backup-timer wait_time
```

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

For example:

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

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

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

Synchronizing a Redundant Configuration

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

There are two types of configuration synchronization:

- **Complete** - On CSSs that have an identical chassis (the same CSS model), produces a running-config on the backup CSS that exactly matches the running-config on the master CSS.
- **Partial** (default) - On CSSs with an incompatible configuration syntax, synchronizes all parameter values in the configuration except the interface and circuit configurations. For example, the master is a CSS 11506 and the backup is a CSS 11150. The script maintains the current backup interface and circuit configurations automatically.

Before You Begin

Before you run the configuration synchronization script, ensure that you have set up the redundancy circuit between the two CSSs and that the Application Peering Protocol (APP) is running on that circuit. For details on configuring the redundancy circuit, see “[Configuring Redundancy](#)” earlier in this chapter. For details on configuring APP, refer to [Chapter 1, Configuring the CSS Domain Name Service](#), in the “[Configuring the Application Peering Protocol](#)” section.

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

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

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

Running the Configuration Synchronization Script

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

```
script play commit_redundancy “arguments”
```

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

```
# commit_RedundConfig “arguments”
```

The arguments for the `commit_redundancy` script are:

- *ip address* - The IP address of the backup CSS. This is the only required argument for this script. For details on automating the entry of the IP address, see “[Setting the BACKUP_IP Variable](#)” later in this section.
- **-a** (All) - Performs a complete configuration synchronization. Use this option only when the master CSS and the backup CSS have identical chassis (see [Table 8-3](#)).
- **-d** (Debug) - Debug switch for the `commit_redundancy` script, which displays the current task being performed as the script progresses. Debug messages display even when you specify the **-s** argument.



Caution

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

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



Note The script verifies the configuration synchronization by default.

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



Note

You can specify the script arguments in any order.

For example:

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

The following output appears:

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

```
Verifying that app session is up with backup switch.
```

```
Making sure app session is up.
```

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

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

```
Making sure app session is up.
```

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

```
Copying running-config to startup-config.
```

```
Archiving startup-config.
```

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

```
Removing CIRCUIT and INTERFACE modes from tmp.cfg.
```

```
Checking if IP redundancy master is set.
```

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

```
Retrieving circuit info for redundancy-protocol link.
```

```
Archiving copy to Backup startup-config.
```

```
Archiving Backup current startup-config.
```

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

```
Clearing running-config.
```

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

```
Making sure app session is down.
```

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

```
Making sure app session is up.
```

```
Comparing the byte count now.
```

```
Config Sync Successful.
```

In this example, the script:

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

For more information on scripts, refer to [Chapter 12, Using the CSS Scripting Language](#).

Config Sync Lock File

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

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

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

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

Now you can run the script again.

Setting the BACKUP_IP Variable

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

To set the BACKUP_IP variable, enter:

```
# set BACKUP_IP "ip_address" {session}
```

where, *ip_address* is the IP address of the backup CSS.

To save the variable in your user profile, enter:

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

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

Logging Configuration Synchronization Script Result Messages

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

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



Note

Log messages are generated with or without the -s (silent) argument specified. See [“Running the Configuration Synchronization Script”](#) earlier in this chapter.

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

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

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

Using the Redundancy Force-Master Command

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

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

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



Note

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

Configuring Multiple Redundant Uplink Services

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

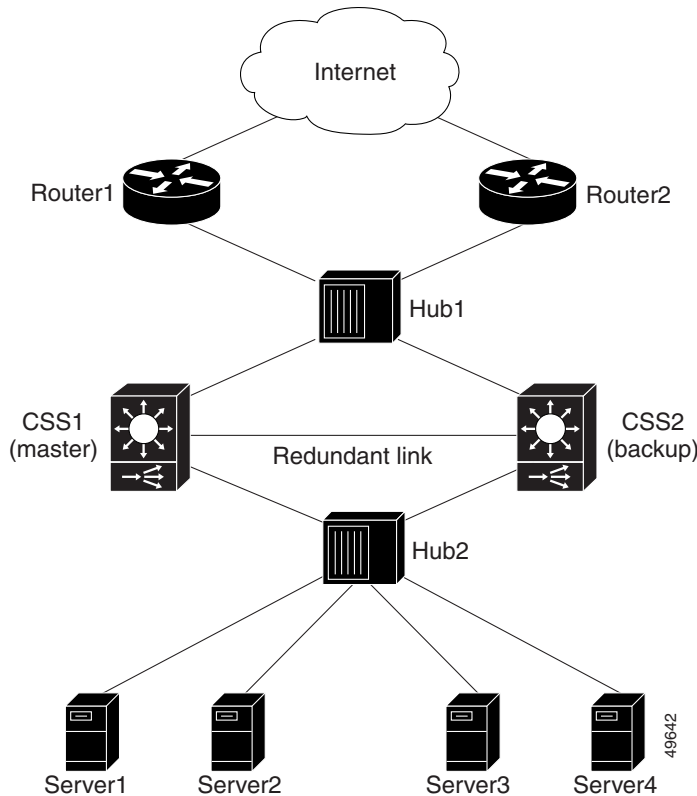
In a redundant configuration that does not configure the routers as type redundancy-up critical services, a backup CSS becomes master only when the current master CSS fails. In this configuration, a switchover *does not* occur when the router services fail.

**Note**

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

Figure 8-2 shows a typical redundant configuration. When CSS1 fails or CSS1 cannot communicate with both the Router1 critical service and the Router2 critical service, CSS2 becomes the master CSS automatically.

Figure 8-2 Multiple Redundant Uplink Services Configuration Example

**Note**

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

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

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

Use the **show redundancy** command to display critical services. See the [“Displaying Redundant Configurations”](#) section.

For example:

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

Using the `redundancy-phy` Command

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



Note

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

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

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



Note

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

Configuring Stateless Redundancy Failover

Use the **redundancy-14-stateless** command in content or group configuration mode to enable stateless redundancy failover in a box-to-box redundancy or a VIP and virtual interface redundancy configuration. Stateless redundancy failover allows critical TCP/IP traffic to continue in case of a failure at the load-balancing CSS by allowing the backup CSS to set up a mid-stream TCP flow. This feature is disabled by default.

The default behavior of a CSS is to set up load-balanced TCP flows only when it receives a TCP frame that begins with SYN. When stateless redundancy failover is enabled and a failover occurs, the backup CSS establishes a mid-stream flow for any existing TCP sessions. The CSS still exhibits the default behavior for all new flows. To restore the default behavior of the CSS for all flows after issuing the **redundancy-14-stateless** command, use the **no redundancy-14-stateless** command.

**Note**

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

Before You Begin

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

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

Environmental Considerations

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

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

**Note**

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

General Configuration Requirements

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

Configuration Restrictions

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

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

Configuring CSS Parameters

The following parameters must match exactly on both redundant CSSs:

- **Stateless redundancy failover command** - Include the **redundancy-l4-stateless** command in both the content rules and the source groups associated with the redundant VIP.
- **Content rules** - Create identical content rules on both CSSs with the following parameters. Refer to the *Cisco Content Services Switch Basic Configuration Guide*.
 - **VIP** - Assign a virtual IP address to each content rule. No wildcard addresses are allowed and no VIP ranges on the content rule are allowed.
 - **Load-balancing method** - Configure the load-balancing method as source IP address (srcip), the only load-balancing method that is supported by stateless redundancy failover.
 - **Failover method** - Configure either 'linear' (default) or 'next' type as the service failover method. 'Bypass' is not supported.
- **Services** - For each load-balanced server farm, configure the following service-related parameters to be the same on both CSSs for each content rule. Refer to the *Cisco Content Services Switch Basic Configuration Guide*.
 - **Service name** - Use identical service names on the master and the backup CSS. Service names are case-sensitive.
 - **IP address** - Use identical IP addresses on the master and the backup CSS.

**Note**

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

- **Service number and order** - The CSS orders services internally in alphabetical order regardless of the order in which you enter them in the configuration.
- **Keepalives** - Create keepalives using the global **keepalive** command, then associate the services with the keepalives using the **keepalive type named** command. Both CSSs must be able to send and receive keepalive messages with the same servers. This helps to ensure that a redistribution of the balance method does not occur in a failover event.

- **Weight** - Routed paths to the load-balanced VIP should be weighted to ensure that a single path is preferred for the lifetime of a TCP/IP connection.
- **Source groups** - Create a source group with the same VIP as the content rule VIP on each CSS to NAT source addresses for packets returning from the server. In case of a failover, the source group will handle the connection setup for TCP/IP transmissions that arrive at the CSS from the servers. All servers in the farm must be members of the configured source group. Refer to the *Cisco Content Services Switch Basic Configuration Guide*.

**Note**

Do not configure source groups for outbound traffic from the servers, because the backup CSS does not know which ports were NATed by the source group on the master CSS if a failure occurs at the master CSS.

Synchronizing the CSS Configurations

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

Box-to-Box Redundancy Configuration

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

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

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

**Note**

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

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

Layer 2 and Layer 3 Configuration and Convergence

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

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

Configuration Example

The following example configuration (see [Figure 8-3](#)) assumes that the:

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

```
ip route 0.0.0.0.0.0.0.0.192.168.20.100
ip redundancy

interface e2
  bridge vlan 1
  description "uplink VLAN"
```

```
interface e5
  bridge vlan 1
  description "uplink VLAN"

interface e9
  bridge vlan 3
  description "server VLAN"

interface e12
  bridge vlan 3
  description "server VLAN"

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

circuit VLAN3
  redundancy
  ip address 192.168.10.1 255.0.0.0

circuit VLAN2
  redundancy-protocol
  ip address 172.7.6.1 255.255.255.253

service s1
  ip address 192.168.10.30
  active

service s2
  ip address 192.168.10.31
  active

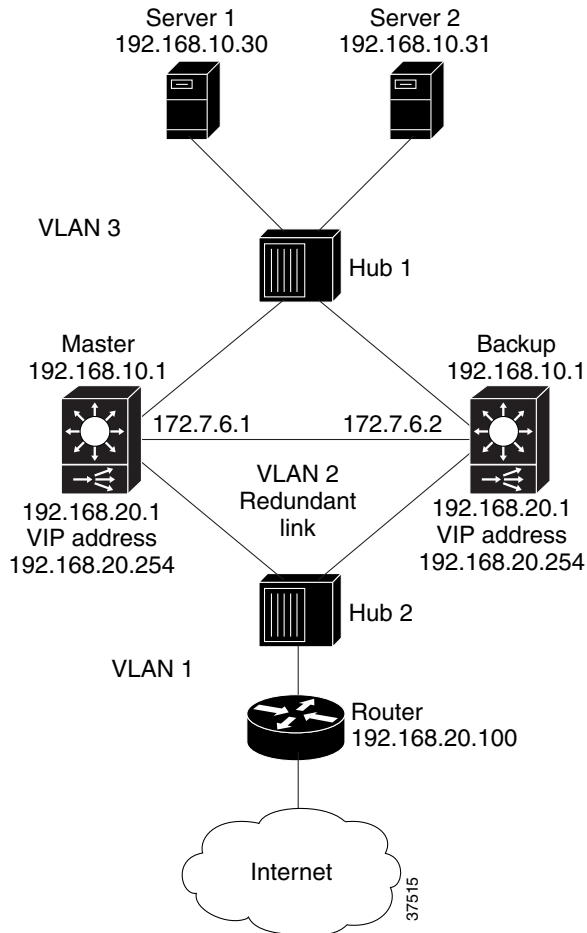
owner Redundant-Pool
  content web
  vip address 192.168.20.254
  protocol tcp
  port 80
  redundancy-l4-stateless
  add s1
  add s2
  balance srcip
  active
```

```

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

```

Figure 8-3 Example Box-to-Box Redundancy Configuration for Stateless Redundancy Failover



VIP and Interface Redundancy Configuration

For details on VIP and virtual interface redundancy, refer to [Chapter 6, Configuring VIP and Virtual Interface Redundancy](#).

Layer 2 and Layer 3 Configuration and Convergence

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

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

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

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

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

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

Configuration Example

The following example configuration (see [Figure 8-4](#)) assumes that the:

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

```
ip route 0.0.0.0 0.0.0.0 192.168.20.100

interface e2
  bridge vlan 1
  description "uplink VLAN"

interface e5
  bridge vlan 1
  description "uplink VLAN"

interface e9
  bridge vlan 3
  description "server VLAN"

interface e12
  bridge vlan 3
  description "server VLAN"

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

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

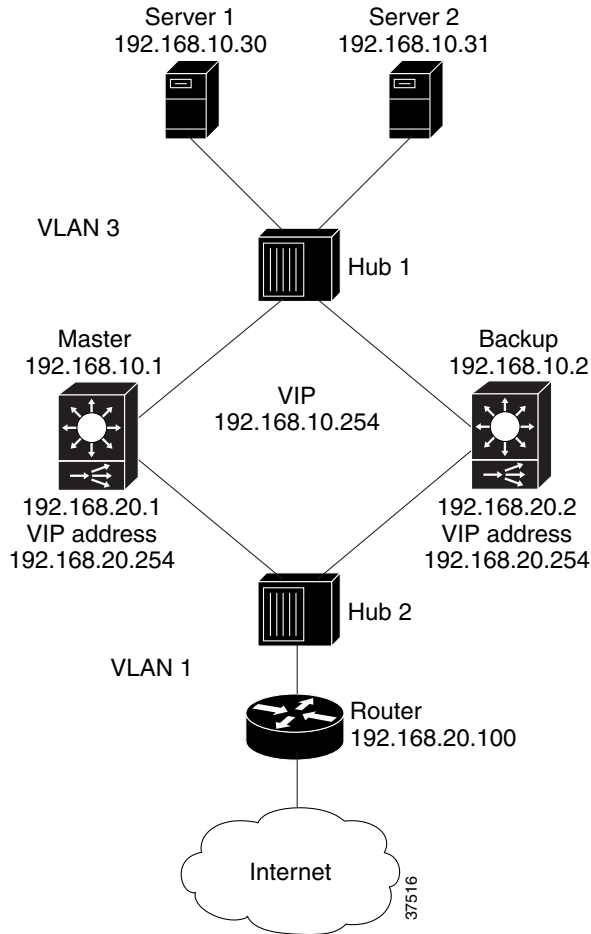
```
service uplink
  ip address 192.168.20.100
  type redundancy-up
  active

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

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

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

Figure 8-4 Example of VIP and Virtual Interface Redundancy Configuration for Stateless Redundancy Failover



Alternative Configurations

Stateless redundancy failover allows other possible configurations and topologies. To use this feature in other high-availability environments, see the other sections in this chapter and to [Chapter 6, Configuring VIP and Virtual Interface Redundancy](#), for details and examples of CSS redundancy configurations. Refer to RFC-2338 *Virtual Router Redundancy Protocol* for additional information.

Managing Your Configuration

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

Other Considerations

The following conditions apply to stateless redundancy failover:

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

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

Displaying Redundant Configurations

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

For example:

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

When redundancy is not configured, the CSS displays the following status:

```
(config)# show redundancy  
Redundancy: Disabled Redundancy Protocol: Not Running
```

The output of the **show redundancy** command varies depending on whether you issue the command on the master or the backup CSS.

Table 8-3 describes the fields in the **show redundancy** output.

Table 8-3 Field Descriptions for the show redundancy Command

Field	Description
Redundancy	Indicates whether or not redundancy is enabled on the CSS.
Redundancy Protocol	Indicates whether or not the redundancy protocol is running on the CSS.
Redundancy State	The current redundancy state of the CSS (Master or Backup).
MasterMode	Indicates whether the CSS is configured as master. Yes indicates that the CSS is designated as master through the ip redundancy master command. No indicates that the CSS is not the designated master through the ip redundancy command.
Number of times redundancy state changed to Master/Backup	The number of times that the CSS has changed to master or backup.
Redundancy interface	The address for the redundancy interface.
Current State Duration	How long the CSS has been in its current redundancy state (master or backup).
Last Fail Reason	A description of the last CSS redundancy failure.
VRID	The virtual router identifier (VRID).
Priority	The priority for the virtual router with its peer. The default priority value is 100. Enter an integer between 1 and 255.
Physical Link Failure Monitor on	
Interface/State	The list of interfaces configured through the redundancy-phy command and their states. The show output is sorted numerically by interface port number.
Uplink Enabled	The number of enabled service uplinks.

Table 8-3 *Field Descriptions for the show redundancy Command (continued)*

Field	Description
Number Alive	The number of alive (Up state) service uplinks.
Service Name/State	The list of uplink services and their states. The show output is sorted numerically by service index number.

■ **Displaying Redundant Configurations**