CHAPTER 16

Configuring Additional Network Interfaces and Bandwidth on Standalone Content Engines

This chapter describes how to set up additional network interfaces and configure bandwidth for these interfaces and content services in a locally managed deployment.

This chapter contains the following sections:

Note For complete syntax and usage information for the CLI commands used in this chapter, see the *Cisco ACNS Software Command Reference, Release 5.5* publication.

Configuring Additional Network Interfaces

When you initially configured a standalone Content Engine, you chose an initial interface and either configured it for DHCP, or gave it a static IP address. You can use the Content Engine CLI to configure additional network interfaces on the Content Engine for redundancy, load balancing, and performance optimization.

You can configure multiple network interfaces as either active-active interfaces or as active-standby interfaces. To configure multiple interfaces as active-active, use the **interface** global configuration command and assign an IP address to each interface. When multiple interfaces are configured on a Content Engine, they are active simultaneously and improve performance.

This section describes how to configure additional interfaces on standalone Content Engines.

About Virtual Interfaces

Port channel and standby group are two types of virtual network interfaces that are supported by standalone Content Engines. For more information, see the following sections:

- Configuring Port Channels (EtherChannel), page 16-2
- Configuring Standby Groups, page 16-4

Configuring Port Channels (EtherChannel)

EtherChannel for the ACNS 5.x software supports the grouping of up to four same-speed network interfaces into one virtual interface. This grouping is called a *port channel* or *channel group*. This grouping capability allows the setting or removing of a virtual interface that consists of two, three, or four Fast Ethernet interfaces or two Gigabit Ethernet interfaces.

EtherChannel also provides interoperability with Cisco routers, switches, and other networking devices or hosts supporting EtherChannel; load balancing; and automatic failure detection and recovery based on each interface's current link status.

To create up to two port channels on a standalone Content Engine, use the **interface PortChannel** configuration command.

Table 16-1 lists the **interface PortChannel** command options.

Table 16-1 Parameters of the interface Portchannel CLI Command

Parameter	Description
PortChannel	Selects Ethernet channel of interfaces to configure.
1	Sets the Port Channel interface number to 1.
2	Sets the Port Channel interface number to 2.
ip	(Optional) Enables IP configuration commands for the interface.
access-group	Configures access control for IP packets on this interface using access control lists (ACLs). For more information about ACLs, see Chapter 19, "Creating and Managing IP Access Control Lists for Standalone Content Engines."
ip-access list	Numeric identifier that identifies the ACL to apply to the current interface. For standard ACLs, the valid range is 1 to 99; for extended ACLs, the valid range is 100 to 199.
accesslist_name	Alphanumeric identifier of up to 30 characters, beginning with a letter that identifies the ACL to apply to the current interface.
in	Applies the specified ACL to inbound packets on the current interface.
out	Applies the specified ACL to outbound packets on the current interface.
address	Sets the interface IP address.
ip-address	IP address of this interface.
netmask	Netmask of this interface.
shutdown	(Optional) Shuts down this interface.

The following example shows how to create a port channel on a Content Engine. In this example, the port channel is port channel 2 and is assigned an IP address of 10.10.10.10 and a netmask of 255.0.0.0.

```
ContentEngine# configure
ContentEngine(config)# interface PortChannel 2
ContentEngine(config-if)# ip address 10.10.10.10 255.0.0.0
ContentEngine(config-if)# exit
```

To remove (tear down) a port channel, use the **no** form of the command, as follows:

```
ContentEngine(config)# interface PortChannel 2
ContentEngine(config-if)# no ip address 10.10.10.10 255.0.0.0
ContentEngine(config-if)# exit
ContentEngine(config)# no interface PortChannel 2
```

To add or remove ports from a port channel, use the commands in the following examples. You can use either the Fast Ethernet or the Gigabit Ethernet ports to form a port channel; however, a port channel cannot contain both Fast Ethernet and Gigabit Ethernet interfaces. A physical interface can be added to a port channel depending upon the device configuration.

To add an interface to a port channel:

```
ContentEngine# configure
ContentEngine(config)# interface FastEthernet 1/1
ContentEngine(config-if)# channel-group 2
ContentEngine(config-if)# exit
```

To remove an interface from a port channel:

```
ContentEngine(config)# interface FastEthernet 1/1
ContentEngine(config-if)# no channel-group 2
ContentEngine(config-if)# exit
```

To configure load balancing for a port channel, use the **port-channel load-balance** global configuration command:

```
ContentEngine(config)# port-channel load-balance
```

The following **load-balance** options are available:

```
dst-ip       Destination IP Address
dst-mac      Destination MAC Address
round-robin  Round robin each interface (default)
```

Round robin allows traffic to be distributed evenly between all interfaces in the port channel. The other balancing options give you flexibility in choosing interfaces when sending an Ethernet frame. The **load-balance** command is effective globally. If two port channels are configured, they have to use the same load-balancing option.

Configuring Standby Groups

In the ACNS 5.2.1 software release and later releases, you can configure one or more interfaces to act as a backup interface (a standby interface) for another interface on a standalone Content Engine. This feature is called standby interface support. Standby groups, which are logical groups of interfaces, are used to implement this feature. When an active network interface fails (because of cable trouble, Layer 2 switch failure, high error count, or other failures) and that interface is part of a standby group, a standby interface can become active and take the load off the failed interface.

There must be at least two interfaces in a standby group. Interfaces that are part of a standby group are called *member interfaces*. After you create a standby group, you define which interfaces should be assigned to this logical group. When defining the member interfaces, you specify the priority of each member interface in a standby group. The member interface with the highest assigned priority is the active interface for that particular standby group. If the active interface fails, the operational member interface with the next highest priority in the standby group comes up, and so forth. If all member interfaces of a particular standby group are down and then one of the member interfaces comes up, the ACNS software detects this situation and brings up the standby group on the member interface that just came up.

In the ACNS 5.3.1 software and later releases, the failure or failover of member interfaces within a standby group will trigger alarms and traps (if alarms and traps are enabled on the Content Engine). Alarms are sent out when failover occurs between member interfaces in a standby group. Specifically, minor alarms are sent out when member interfaces fail, and these alarms are cleared automatically when the interface failover has been successfully completed. Major alarms are sent out if the standby group goes down (that is, no member interface in a standby group can be brought up.)

Note A physical interface can belong to more than one standby group. Consequently, a single interface can act as a standby interface for more than one standby group.

This standby interface feature can also be used to support a redundant network that uses Layer 4 Cisco Content Services Switch [CSS] switches) to load balance requests to multiple Content Engines. The CSS switch supports active-standby configuration. If the active CSS switch fails, the standby CSS switch takes over all of the load. In such a case, the Content Engine detects this failure and starts serving the

same IP address (shared IP address) on the standby network interface card (NIC) and preserves the existing TCP sessions (session-level redundancy). Session-level redundancy is only possible if the CSS switch can preserve sessions in a failure situation. If the CSS switch loses the sessions, the session will be lost.

To configure standby interfaces, interfaces are logically assigned to standby groups. The following rules define the standby group relationships:

- Each standby group is assigned a unique standby IP address, shared by all member interfaces of the standby group. The IP address of the standby group is shared among the member interfaces; however, only the active interface of the standby group uses this shared IP address at any one time. This shared IP address is configured as an alias on the active interface.
- In the ACNS 5.2.x software release, a physical interface needed a dummy or valid IP address assigned to it before you could add the physical interface to a standby group. In the ACNS 5.3.1 software and later releases, this is no longer a requirement.
- Configure the duplex and speed settings of the member interfaces for better reliability.
- If the active interface fails, the operational interface in its standby group that is assigned the next highest priority becomes active. However, when the interface with the higher priority recovers, it does not become active again without manual intervention.
- If all the member interfaces of a standby group fail and then one recovers, the ACNS software brings up the standby group on the operational member interface.
- If a physical interface is a member of a port channel group, it cannot join a standby group. Likewise, if a physical interface is a member of a standby group, it cannot join a port channel group.

Note Interface IP addresses and standby group IP addresses must be on different subnets to ensure reliable operation. Make sure to configure the interface default gateway using the **ip default-gateway** global configuration command instead of the **ip route** command.

To create standby groups on standalone Content Engines, use the **interface standby** global configuration command. In the ACNS 5.3.1 software release, the CLI syntax for configuring standby groups was changed to make it more similar to the port channel CLI syntax.

Table 16-2 lists the **interface standby** command options.

Table 16-2 Parameters of the interface Standby CLI Command

Parameter	Description
standby	Selects the standby group of interfaces to configure.
1	Specifies Standby Group 1.
2	Specifies Standby Group 2.
3	Specifies Standby Group 3.
4	Specifies Standby Group 4.
priority	Sets the priority of the member interface within a standby group. The priority of a member interface can be changed at runtime. The member interface that has the highest priority after this change becomes the new active interface (the default action is to preempt the currently active interface if an interface with higher priority exists).

Table 16-2 Parameters of the interface Standby CLI Command (continued)

Parameter	Description
priority number	Each member interface is assigned a priority number. The member interface with the highest priority number is the active interface for that standby group. Only the active interface uses the group IP address. If the **priority** option is specified without a priority number, the default value of 100 is used.
ip address	Sets the IP address for the specified standby group (Standby Group 1, 2, 3, or 4).
ip-address	IP address of the specified standby group (Standby Group 1, 2, 3, or 4). The group IP address and netmask of a standby group must be configured on all of the member interfaces.
netmask	Netmask of the specified standby group (Standby Group 1, 2, 3, or 4).
errors	Sets the maximum number of errors allowed on the active interface before the interface is shut down and the standby interface is brought up. This option is disabled by default.
max-error-number	Specifies the maximum number of transmit and receive errors that are allowed on the active interface before the interface is shut down and the standby interface is brought up. The value can be from 0 to 4294967295.
shutdown	(Optional) Shuts down the specified standby group (Standby Group 1, 2, 3, or 4). You can shut down a standby group even if you have not configured a group IP address for the standby group.

Note Unlike port channels, standby groups do not support IP ACLs at a group level. However, you can configure a member interface of a standby group to support an IP ACL at the interface level. For example, you can individually configure the two member interfaces of Standby Group 1 (the Fast Ethernet slot 0/port 0 interface and the Fast Ethernet slot 0/port 1 interface) to support an IP ACL named ACL1 but you cannot configure the Standby Group 1 to support ACL1.

The following example shows how to create a standby group on a standalone Content Engine, and then add and remove members from this standby group:

Step 1 Create a standby group. In this case, the standby group is Standby Group 1.

```
ContentEngine# configure
ContentEngine(config)# interface standby 1
ContentEngine(config-if)#
```

Step 2 Assign a group IP address and netmask to Standby Group 1.

In ACNS 5.3.1 software and later releases, you can configure a group IP address regardless of whether the standby group is shut down or not.

In this example, Standby Group 1 is assigned a group IP address of 10.10.10.10 and a netmask of 255.0.0.0:

```
ContentEngine(config-if)# ip address 10.10.10.10 255.0.0.0
ContentEngine(config-if)# errors 500
```

Step 3 Define the member interfaces of Standby Group 1. Specify the priority of each member interface that is added to Standby Group 1.

The following example shows how to add two Fast Ethernet interfaces to Standby Group 1, and then assign each of these member interfaces a priority within the group:

a. A FastEthernet interface (slot 0/port 0) is added to Standby Group 1 and assigned a priority of 150.

```
ContentEngine(config)# interface FastEthernet 0/0
ContentEngine(config-if)# standby 1 priority 150
```

b. A second FastEthernet interface (slot 0/port 1) is added to Standby Group 1 and assigned a priority of 100 (the default value).

```
ContentEngine(config)# interface FastEthernet 0/1
ContentEngine(config-if)# standby 1
ContentEngine(config-if)# exit
ContentEngine(config)#
```

Because FastEthernet 0/0 is assigned the highest priority (a priority number of 150) of all the member interfaces in the group, it will be chosen as the active interface for the group if it can be brought up.

Step 4 Specify the maximum number of transmit and receive errors that should be allowed on the active interface before the interface is shut down and the standby interface is brought up.

In this case, the maximum number of errors is set to 500.

```
ContentEngine(config-if)# errors 500
ContentEngine(config-if)# exit
ContentEngine(config)#
```

Step 5 Display information about the standby group configuration by entering the **show standby** EXEC command.

In the following sample command output, one standby group (Standby Group 1) is configured on this Content Engine. The command output also shows which member interface is the active interface. In this case, the active interface is the Fast Ethernet slot 0/port 0 interface.

```
CE-560# show standby
Standby Group: 1
          IP address: 10.10.10.10,   netmask: 255.255.255.0
          Member interface:
                        FastEthernet 0/0              priority: 150
                        FastEthernet 0/1              priority: 100
           Active interface: FastEthernet 0/0
CE-560#
```

Note To display information about a specific standby group configuration, enter the **show interface standby** *standby group number* EXEC command.

Step 6 To remove a member interface (for example, the Fast Ethernet slot 0/port 1 interface) from Standby Group 1, use the **no** form of the **standby** command:

```
ContentEngine(config)# interface FastEthernet 0/1
ContentEngine(config-if)# no standby 1
ContentEngine(config-if)# exit
ContentEngine(config)#
```

Step 7 To shut down a standby group, use the **interface standby** command to specify the group that you want to shut down and then enter the **shutdown** command to shut it down.

When a standby group is shut down, all of the alarms previously raised by this standby group are cleared.

The following example shows how to shut down Standby Group 1:

```
ContentEngine(config)# interface standby 1
ContentEngine(config-if)# exit
ContentEngine(config)# exit
```

Step 8 To tear down a standby group, use the **no** form of the **interface standby** command.

The following example shows how to tear down Standby Group 1:

```
ContentEngine(config)# interface standby 1
ContentEngine(config-if)# no ip address 10.10.10.10 255.0.0.0
Please remove member interface(s) from this standby group first.
ContentEngine(config)# interface GigabitEthernet 2/0
ContentEngine(config-if)# no standby 1
ContentEngine(config-if)# exit
ContentEngine(config)# interface standby 1
ContentEngine(config-if)# no ip address 10.10.10.10 255.0.0.0
ContentEngine(config-if)# exit
ContentEngine(config)# no interface standby 1
ContentEngine(config)# exit
```

Specifying the Primary Interface

To specify the primary interface for the standalone Content Engine, use the **primary-interface** global configuration command.

primary-interface {**FastEthernet** *0-3/port* | **GigabitEthernet** *1-2/port* | **PortChannel** *1-2* | **Standby** *group_num*}

The following example shows how to specify the Fast Ethernet slot 0 port 0 as the primary interface on a Content Engine model CE-7320:

```
CE-7320(config)# primary-interface FastEthernet 0/0
```

To change the primary interface, reenter the command string and specify a different interface.

In the ACNS 5.2.1 software and later releases, you can select a standby interface as the primary interface (that is, you can enter the **primary-interface standby** *group-num* command to specify a standby group as the primary interface on a standalone Content Engine).

If you use the **restore factory-default preserve basic-config** command, the configuration for the primary interface is not preserved. On an ACNS 5.x device, if you want to reenable the ACNS network after using the **restore factory-default preserve basic-config** command, make sure to reconfigure the primary interface after the factory defaults are restored.

Configuring Multiple Secondary IP Addresses on a Single Physical Interface

You can assign up to four secondary IP addresses to a single physical interface on a Content Engine. By configuring multiple IP addresses on a single interface, the Content Engine can be present in more than one subnet. This allows you to optimize response time because the content goes directly from the Content Engine to the requesting client without being redirected through a router. The Content Engine becomes visible to the client because both are configured on the same subnet.

If a Content Engine has one physical interface that has multiple secondary IP addresses assigned to it, the egress traffic uses the source IP address that is chosen by IP routing. If the secondary IP addresses are in the same subnet as the primary IP address, then the egress traffic only uses the primary IP address. In contrast, if the secondary IP addresses are in a different subnet than the primary IP address, then the destination IP address determines which IP address on the Content Engine is used for the egress traffic.

To set these secondary IP addresses on a standalone Content Engine, use the **ip address** configuration interface command:

```
ContentEngine# configure
ContentEngine(config)# interface FastEthernet 0/0
ContentEngine(config-if)# ip address 10.10.10.10 255.0.0.0 secondary
```

These secondary IP addresses become active only after the primary IP address is configured. No two interfaces can have the same IP address in the same subnet.

Configuring the Fibre Channel Interface

The ACNS 5.x software supports Fibre Channel interfaces. Fibre Channel is the chosen technology for interconnecting storage devices and servers in a storage area network (SAN). In a SAN, the storage does not need to be directly attached to the server, and data transfer happens over a high-throughput, high-availability network. Fibre Channel can operate at speeds of 1 gigabit per second (Gbps) and 2 Gbps.

To detect the presence of Fibre Channel storage, the storage array must be configured to assign storage space for the Content Engine, and the Content Engine must be reloaded before it can detect the storage assignment. To confirm whether the Content Engine has detected the storage assignment, use the **show disks** and the **show disks details** EXEC commands.

To configure the Fibre Channel interface on the Content Engine, use the **interface FibreChannel** *slot/port* interface configuration command. For example:

```
ContentEngine# configure
ContentEngine(config)# interface FibreChannel 0/0
ContentEngine(config-if)#?
  exit   Exit from this submode
  mode   Change the fibre channel interface operating mode
  no     Negate a command or set its defaults
  speed  Change the fibre channel interface speed
ContentEngine(config-if)# mode ?
  autosense        Use this mode to have the CE autosense
  direct-attached  Use this mode when the CE is directly connected to storage array
  switched         Use this mode when the CE is connected to a switch
ContentEngine(config-if)# speed ?
  1          1Gbps
  2          2Gbps
  autosense  autosense
```

Note For a complete description of the **interface FibreChannel** command syntax and usage, see the *Cisco ACNS Software Command Reference, Release 5.5* publication. For information regarding which Fibre Channel storage arrays are supported by Cisco Systems, see the *Release Notes for Cisco ACNS Software, Release 5.5*.

Defining Interface Descriptions

In the ACNS 5.3.1 software and later releases, you can specify a one-line description for a specific interface on a Content Engine. You use the **description** *description text* interface configuration command to enter the description for the specific interface. The maximum length of the description text is 240 characters.

This feature is supported for the following interfaces: FastEthernet, GigabitEthernet, FibreChannel, PortChannel, and Standby. This example shows how to enter an interface description:

```
ContentEngine(config)# interface FastEthernet 0/0
ContentEngine(config-if)# description "This an interface to the WAN."
```

Note This feature is not currently supported for the SCSI or IDE interfaces

After you define the description for an interface, use the **show** EXEC commands to display the defined interface descriptions. For example, to display all of the defined interface descriptions enter this command:

```
ContentEngine# show running-config
.
.
.
interface FastEthernet 0/0
 description This is interface to WAN
 ip address dhcp
 ip address 192.168.1.200 255.255.255.0
 no autosense
 bandwidth 100
 full-duplex
 exit
.
.
.
```

To display the defined description for a specific interface on the Content Engine, enter the **show interface** *interface type slot/port* EXEC command. As the following excerpt of the command output shows, the description of the specified interface is displayed as the first line in the command line output:

```
ContentEngine# show interface GigabitEthernet 1/0
DescriptionN This is the interface to the lab
type: Ethernet
.
.
.
```

Configuring Bandwidth for Interfaces and Content Services

With the various types of traffic originating from a device, every type of traffic, such as streaming media, HTTP, and metadata, consumes network resources.

Configuring Interface Bandwidth

To configure an interface bandwidth on a standalone Content Engine, use the **bandwidth** interface configuration command. Bandwidth is specified in megabits per second (Mbps). The **1000** Mbps option is not available on all ports and is the same as autosense.

bandwidth {**10** | **100** | **1000**}

no bandwidth {**10** | **100** | **1000**}

To restore default values, use the **no** form of this command.

For a Content Engine CE-7320 model that has an optical Gigabit Ethernet interface the speed of this interface cannot be changed. Therefore, Gigabit Ethernet interfaces only run at 1000 Mbps for a CE-7320. For newer models of the Content Engine (for example, the CE-510, CE-511, CE-565, CE-566, CE-7305, CE-7325, and CE-7326) that have a Gigabit Ethernet interface over copper, this restriction does not apply; you can configure these Gigabit Ethernet interfaces to run at 10, 100, or 1000 Mbps. On these newer Content Engine models, the 1000 Mbps setting implies autosense (for example, you cannot configure the Gigabit Ethernet interface to run at 1000 Mbps and half duplex). The ACNS 5.x software automatically enables autosense if the speed is set to 1000 Mbps.

In the ACNS 5.3.1 software and later releases, you can configure the Gigabit Ethernet interface settings (autosense, bandwidth, and duplex settings) if the Gigabit-over-copper interface is up or down. If the interface is up, it will apply the specific interface settings. If the interface is down, the specified settings are stored and then applied when the interface is brought up. For example, you can specify any of the following commands for an Gigabit-over-copper interface, which is currently down, and have these settings automatically applied when the interface is brought up:

```
ContentEngine(config-if)# bandwidth 10
ContentEngine(config-if)# bandwidth 100
ContentEngine(config-if)# bandwidth 1000
ContentEngine(config-if)# autosense
ContentEngine(config-if)# half-duplex
ContentEngine(config-if)# full-duplex
```

Note In the ACNS 5.2.x software and earlier releases, you could only configure the Gigabit Ethernet interface settings if the interface is up.

You can not configure the Gibabit Ethernet interface settings if it is an optical Gigabit Ethernet interface (for example, if the Content Engine is a CE-7320 model).

With the ACNS 5.x software, you can also configure a maximum bandwidth for the preloading process using the **pre-load max-bandwidth** global configuration command.

Configuring Bandwidth for Content Services

To specify bandwidth limits for WMT and RealProxy live content or the streaming media that is being cached on the standalone Content Engine, use the **bandwidth** global configuration command.

For each type of content service (WMT and RealProxy), you can specify the maximum amount of bandwidth on the Content Engine that should be allocated to that service during a specified period. This is called scheduled bandwidth. For example, you can limit the RealProxy bandwidth to 1000 kbps from Monday at 8:00 a.m. to Friday at 6:00 p.m.

```
ContentEngine(config)# bandwidth allow 1000 real-proxy start-time monday 8:00
end-time friday 18:00
```

Although there are no default values for any of the bandwidth configuration fields, the values that you enter depend upon the bandwidth license in effect for your specific system. If you enter a value that is beyond the allowable bandwidth based on your system's bandwidth capacity, the value is accepted but a warning message is displayed. Internally, your system bandwidth is limited to the maximum value granted by the license. All values entered are in kilobits per second (kbps).

For information about how to configure incoming and outgoing WMT bandwidth and bit rates on standalone Content Engines, see the "Configuring Incoming and Outgoing WMT Bandwidth and Bit Rates" section on page 9-23.

In the ACNS 5.3.1 software and later releases, you can configure IP subnet-based bandwidth control for WMT requests. For more information on this topic, see the "Configuring Subnet-Based Outgoing Bandwidth" section on page 9-24.