Configuring Ethernet, Fast Ethernet, and Gigabit Ethernet Switching

This chapter describes how to use the command-line interface (CLI) to configure Ethernet, Fast Ethernet, and Gigabit Ethernet switching on the Catalyst 6000 family switches. The configuration tasks in this chapter apply to Ethernet, Fast Ethernet, and Gigabit Ethernet switching modules, as well as to the uplink ports on the supervisor engine.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6000 Family Command Reference* publication.

This chapter consists of these sections:

Understanding How Ethernet Works

Catalyst 6000 family switches support simultaneous, parallel connections between Ethernet segments. Switched connections between Ethernet segments last only for the duration of the packet. New connections can be made between different segments for the next packet.

Catalyst 6000 family switches solve congestion problems caused by high-bandwidth devices and a large number of users by assigning each device (for example, a server) to its own 10-, 100-, or 1000-Mbps segment. Because each Ethernet port on the switch represents a separate Ethernet segment, servers in a properly configured switched environment achieve full access to the bandwidth.

Because collisions are a major bottleneck in Ethernet networks, an effective solution is full-duplex communication, which is an option for any 10- or 100-Mbps port on a Catalyst 6000 family switch (Gigabit Ethernet ports are always full duplex). Normally, Ethernet operates in half-duplex mode, which means that stations can either receive or transmit. In full-duplex mode, two stations can transmit and receive at the same time. When packets can flow in both directions simultaneously, effective Ethernet bandwidth doubles to 20 Mbps for 10-Mbps ports and to 200 Mbps for Fast Ethernet ports. Gigabit Ethernet ports on Catalyst 6000 family switches are full duplex only (2-Gbps effective bandwidth).

These sections describe Ethernet:

Switching Frames Between Segments

Each Ethernet port on a Catalyst 6000 family switch can connect to a single workstation or server, or to a hub through which workstations or servers connect to the network.

Ports on a typical Ethernet hub all connect to a common backplane within the hub, and the bandwidth of the network is shared by all devices attached to the hub. If two stations establish a session that uses a significant level of bandwidth, the network performance of all other stations attached to the hub is degraded.

To reduce degradation, the switch treats each port as an individual segment. When stations on different ports need to communicate, the switch forwards frames from one port to the other at wire speed to ensure that each session receives full bandwidth.

To switch frames between ports efficiently, the switch maintains an address table. When a frame enters the switch, it associates the MAC address of the sending station with the port on which it was received.

Building the Address Table

Catalyst 6000 family switches build the address table by using the source address of the frames received. When the switch receives a frame for a destination address not listed in its address table, it floods the frame to all ports of the same VLAN except the port that received the frame. When the destination station replies, the switch adds its relevant source address and port ID to the address table. The switch then forwards subsequent frames to a single port without flooding to all ports.

The address table can store at least 32K address entries without flooding any entries. The switch uses an aging mechanism, defined by a configurable aging timer, so if an address remains inactive for a specified number of seconds, it is removed from the address table.

Understanding How Port Negotiation Works

Note Port negotiation does not involve negotiating port speed. You cannot disable port negotiation with the **set port speed** command.

Port negotiation exchanges flow-control parameters, remote fault information, and duplex information. Configure port negotiation with the **set port negotiation** command. Port negotiation is enabled by default.

The ports on both ends of a link must have the same setting. The link will not come up if the ports at each end of the link are set inconsistently (port negotiation enabled on one port and disabled on the other).

Table 4-1 shows the four possible port negotiation configurations and the resulting link status for each configuration.

Table 4-1 Port Negotiation Configuration and Possible Link Status

Port Negotiation State		Link Status	
Near End[1]	**Far End**[2]	**Near End**	**Far End**
Off	Off	Up	Up
On	On	Up	Up
Off	On	Up	Down
On	Off	Down	Up

1. Near End refers to the local port.
2. Far End refers to the port at the other end of the link.

Default Ethernet, Fast Ethernet, and Gigabit Ethernet Configuration

Table 4-2 shows the Ethernet, Fast Ethernet, and Gigabit Ethernet default configuration.

Table 4-2 Ethernet Default Configuration

Feature	Default Value
Port enable state	All ports are enabled
Port name	None
Duplex mode	• Half duplex for 10-Mbps Ethernet ports • Autonegotiate speed and duplex for 10/100-Mbps Fast Ethernet ports • Autonegotiate duplex for 100-Mbps Fast Ethernet ports • Full duplex for 1000-Mbps Gigabit Ethernet ports
Flow control (Gigabit Ethernet)	Flow control set to off for receive (Rx) and desired for transmit (Tx)
Flow control (other Ethernet)	Flow control set to off for receive (Rx); transmit (Tx) not supported
Spanning Tree Protocol (STP)	Enabled for VLAN 1
Native VLAN	VLAN 1
Port VLAN cost	• Port VLAN cost of 100 for 10-Mbps Ethernet ports • Port VLAN cost of 19 for 10/100-Mbps Fast Ethernet ports • Port VLAN cost of 19 for 100-Mbps Fast Ethernet ports • Port VLAN cost of 4 for 1000-Mbps Gigabit Ethernet ports
EtherChannel	Disabled on all Ethernet ports
Jumbo frames	Disabled on all Ethernet ports

Setting the Port Configuration

These sections describe how to configure Ethernet, Fast Ethernet, and Gigabit Ethernet switching on the Catalyst 6000 family switches:

Setting the Port Name

You can set port names on Ethernet, Fast Ethernet, and Gigabit Ethernet switching modules to facilitate switch administration.

To set the port name, perform this task in privileged mode:

	Task	Command
Step 1	Set a port name.	**set port name** *mod/port* [*name_string*]
Step 2	Verify the port name is configured.	**show port** [*mod*[*/port*]]

This example shows how to set the name for ports 1/1 and 1/2 and how to verify that the port names are configured correctly:

```
Console> (enable) set port name 1/1 Router Connection
Port 1/1 name set.
Console> (enable) set port name 1/2 Server Link
Port 1/2 name set.
Console> (enable) show port 1
Port  Name               Status     Vlan         Duplex Speed Type
----- ------------------ ---------- ----------   ------ ----- ------------
 1/1  Router Connection  connected  trunk          full  1000 1000BaseSX
 1/2  Server Link        connected  trunk          full  1000 1000BaseSX

<...output truncated...>

Last-Time-Cleared
--------------------------
Wed Jun 16 1999, 16:25:57
Console> (enable)
```

Setting the Port Speed

You can configure the port speed on 10/100-Mbps Ethernet switching modules. Use the **auto** keyword to autonegotiate the port's speed and duplex mode with the neighboring port.

Note If the port speed is set to **auto** on a 10/100-Mbps Ethernet port, both speed and duplex are autonegotiated.

To set the port speed for a 10/100-Mbps port, perform this task in privileged mode:

	Task	Command
Step 1	Set the port speed of a 10/100-Mbps Fast Ethernet port.	**set port speed** *mod/port* {**10** | **100** | **auto**}
Step 2	Verify that the speed of the port is configured correctly.	**show port** [*mod*[*/port*]]

This example shows how to set the port speed to 100 Mbps on port 2/2:

```
Console> (enable) set port speed 2/2 100
Port 2/2 speed set to 100 Mbps.
Console> (enable)
```

This example shows how to make port 2/1 autonegotiate speed and duplex with the neighboring port:

```
Console> (enable) set port speed 2/1 auto
Port 2/1 speed set to auto-sensing mode.
Console> (enable)
```

Setting the Port Duplex Mode

You can set the port duplex mode to full or half duplex for Ethernet and Fast Ethernet ports.

Note Gigabit Ethernet is full duplex only. You cannot change the duplex mode on Gigabit Ethernet ports.

Note If the port speed is set to **auto** on a 10/100-Mbps Ethernet port, both speed and duplex are autonegotiated. You cannot change the duplex mode of autonegotiation ports.

To set the duplex mode of a port, perform this task in privileged mode:

	Task	Command
Step 1	Set the duplex mode of a port.	**set port duplex** *mod/port* {**full** | **half**}
Step 2	Verify that the duplex mode of the port is configured correctly.	**show port** [*mod*[*/port*]]

This example shows how to set the duplex mode to half duplex on port 2/1:

```
Console> (enable) set port duplex 2/1 half
Port 2/1 set to half-duplex.
Console> (enable)
```

Configuring IEEE 802.3X Flow Control

Gigabit Ethernet ports on the Catalyst 6000 family switches use flow control to inhibit the transmission of packets to the port for a period of time; other Ethernet ports use flow control to respond to flow-control requests.

If a Gigabit Ethernet port receive buffer becomes full, the port transmits a "pause" packet that tells remote ports to delay sending more packets for a specified period of time. All Ethernet ports (1000 Mbps, 100 Mbps, and 10 Mbps) can receive and act upon "pause" packets from other devices.

Enter the **set port flow control** command to configure flow control on ports. Table 4-3 lists the **set port flowcontrol** command keywords and describes their functions.

Table 4-3 Ethernet-Flow Control Keyword Functions

Keywords	Function
receive on	The port uses flow control dictated by the neighboring port.
receive desired	The port uses flow control if the neighboring port uses it and does not use flow control if the neighboring port does not use it.
receive off	The port does not use flow control, regardless of whether flow control is requested by the neighboring port.
send on[1]	The port sends flow-control frames to the neighboring port.
send desired[1]	The port sends flow-control frames to the port if the neighboring port asks to use flow control.
send off[1]	The port does not send flow-control frames to the neighboring port.

1. Supported only on Gigabit Ethernet ports.

To configure flow control, perform this task in privileged mode:

	Task	Command
Step 1	Set the flow-control parameters.	**set port flowcontrol** *mod/port* {**receive** | **send**} {**off** | **on** | **desired**}
Step 2	Verify the flow-control configuration.	**show port flowcontrol**

This example shows how to turn transmit and receive flow control on and how to verify the flow-control configuration:

```
Console> (enable) set port flowcontrol 3/1 send on
Port 3/1 will send flowcontrol to far end.
Console> (enable) set port flowcontrol 3/1 receive on
Port 3/1 will require far end to send flow control
```

```
Console> (enable) show port flowcontrol
Port  Send-Flowcontrol  Receive-Flowcntl  RxPause  TxPause
      Admin   Oper      Admin   Oper
----- ----------------  ----------------  -------  -------
 3/1  on      disagree  on      disagree  0        0
 3/2  off     off       off     off       0        0
 3/3  desired on        desired off       10       10
Console> (enable)
```

Enabling and Disabling Port Negotiation

To enable port negotiation, perform this task in privileged mode:

	Task	Command
Step 1	Enable port negotiation.	**set port negotiation** *mod*/*port* **enable**
Step 2	Verify the port negotiation configuration.	**show port negotiation** [*mod*/*port*]

This example shows how to enable port negotiation and verify the configuration:

```
Console> (enable) set port negotiation 2/1 enable
Port 2/1 negotiation enabled
Console> (enable) show port negotiation 2/1
Port   Link Negotiation
-----  ----------------
 2/1   enabled
Console> (enable)
```

To disable port negotiation, perform this task in privileged mode:

	Task	Command
Step 1	Disable port negotiation.	**set port negotiation** *mod*/*port* **disable**
Step 2	Verify the port negotiation configuration.	**show port negotiation** [*mod*/*port*]

This example shows how to disable port negotiation and verify the configuration:

```
Console> (enable) set port negotiation 2/1 disable
Port 2/1 negotiation disabled
Console> (enable) show port negotiation 2/1
Port   Link Negotiation
-----  ----------------
 2/1   disabled
Console> (enable)
```

Changing the Default Port Enable State

Note Changing the default port enable state applies to all port types, not just Ethernet.

Note This feature is not supported on systems that do not have a chassis ID PROM.

When you enter the **clear config all** command or in the event of a configuration loss, all ports collapse into VLAN 1. This might cause a security and network instability problem. Entering the **set default portstatus** command puts all ports into a disable state and blocks the traffic flowing through the ports during a configuration loss. You can then manually configure the ports back to the enable state.

The default port status configuration is stored on the chassis. This means it is tied to a chassis and not the supervisor engine. The **clear config all** command uses this setting to determine whether ports should be enabled or disabled when returning to default configuration. The **clear config all** command does not change the default port status setting on the chassis. The output of the **show config** command shows the current default port status configuration.

To change the port enable state, perform this task in privileged mode:

	Task	Command
Step 1	Change the port enable state.	**set default portstatus** {**enable** | **disable**}
Step 2	Display the port enable state.	**show default**

This example shows how to change the default port enable state from enabled to disabled:

```
Console> (enable) set default portstatus disable
Default port status set to disable.
Console> (enable)
```

This example shows how to display the port enable state:

```
Console> (enable) show default
portstatus: disable
Console> (enable)
```

Setting the Port Debounce Timer

You can set the port debounce timer on a per-port basis for Ethernet, Fast Ethernet, and Gigabit Ethernet ports. When you set the port debounce timer, the switch delays notifying the main processor of a link change that can decrease traffic loss due to a network outage.

Caution Enabling the port debounce timer causes link up and link down detections to be delayed, resulting in loss of data traffic during the debouncing period. This situation might affect the convergence and reconvergence of various Layer 2 and Layer 3 protocols.

Table 4-4 lists the time delay that occurs before the switch notifies the main processor of a link change before and after the switch enables the debounce timer.

Table 4-4 Port Debounce Timer Delay Time

Port Type	Debounce Timer Disabled	Debounce Timer Enabled
10BASE-FL ports	300 milliseconds	3100 milliseconds
10/100BASE-TX ports	300 milliseconds	3100 milliseconds
100BASE-FX ports	300 milliseconds	3100 milliseconds
10/100/1000BASE-TX ports	300 milliseconds	3100 milliseconds
1000BASE-TX ports	300 milliseconds	3100 milliseconds
Fiber Gigabit Ethernet ports	10 milliseconds	100 milliseconds

To set the debounce timer on a port, perform this task in privileged mode:

	Task	Command
Step 1	Enable the debounce timer for a port.	**set port debounce** *mod num/port num* {**enable** | **disable**}
Step 2	Verify that the debounce timer of the port is configured correctly.	**show port debounce** [*mod* | *mod_num/port_num*]

This example shows how to enable the debounce timer on port 2/1:

```
Console> (enable) set port debounce 2/1 enable
Link debounce enabled on port 2/1
Console> (enable)
```

This example shows how to display the per-port debounce timer settings:

```
Console> (enable) show port debounce
Port  Link debounce
-----  ---------------
 2/1   enable
 2/2   disable
Console> (enable)
```

Configuring a Timeout Period for Ports in errdisable State

A port is in errdisable state if it is enabled in NVRAM, but is disabled at runtime by any process. For example, if UniDirectional Link Detection (UDLD) detects a unidirectional link, the port shuts down at runtime. However, because the NVRAM configuration for the port is enabled (you have not disabled the port), the port status is shown as errdisable.

Once a port is in the errdisable state, you have to reenable it manually. The errdisable timeout feature allows you to configure a timeout period for ports in errdisable state; the ports are reenabled automatically eliminating the need to reenable all the errdisabled ports manually.

A port enters errdisable state for the following reasons (these reasons appear as configuration options with the **set errdisable-timeout enable** command):

- Channel misconfiguration
- Duplex mismatch
- BPDU port-guard
- UDLD
- Other (reasons other than the above)
- All (apply errdisable timeout to all reasons)

You can enable or disable errdisable timeout for each of the above listed reasons. The ports in errdisable state for reasons other than the first four reasons are considered "other." If you specify "other," all ports errdisabled by causes *other* than the first four reasons are enabled for errdisable timeout. If you specify "all," all ports errdisabled for any reason are enabled for errdisable timeout.

The errdisable feature is disabled by default. The default interval for enabling a port is 300 seconds. The allowable interval range is 30 to 86400 seconds (30 seconds to 24 hours).

This example shows how to enable errdisable timeout for BPDU guard causes:

```
Console> (enable) set errdisable-timeout enable bpdu-guard
Successfully enabled errdisable-timeout for bpdu-guard.
Console> (enable)
```

This example shows how to enable errdisable timeout for all causes:

```
Console> (enable) set errdisable-timeout enable all
Successfully enabled errdisable-timeout for all.
Console> (enable)
```

This example shows how to set the errdisable timeout interval to 450 seconds:

```
Console> (enable) set errdisable-timeout interval 450
Successfully set errdisable timeout to 450 seconds.
Console> (enable)
```

This example shows how to display the errdisable timeout configuration:

```
Console> (enable) show errdisable-timeout
ErrDisable Reason   Timeout Status
------------------- ------------
bpdu-guard          Enable
channel-misconfig   Disable
duplex-mismatch     Enable
udld                Enable
other               Disable

Interval: 300 seconds

Ports that will be enabled at the next timeout:
Port  ErrDisable Reason
----- -----------------
 3/1  udld
 3/8  bpdu-guard
 6/5  udld
 7/24 duplex-mismatch
Console> (enable)
```

Configuring the Jumbo Frame Feature

These sections describe the jumbo frame feature:

Configuring the Jumbo Frame Feature on the Supervisor Engine

When you enable the jumbo frame feature on a port, the port can switch large (or *jumbo*) frames. This feature is useful in optimizing server-to-server performance. The default maximum transmission unit (MTU) frame size is 1548 bytes for all Ethernet ports. By enabling the jumbo frame feature on a port, the MTU size is increased to 9216 bytes.

To enable the jumbo frame feature on a per-port basis, follow these guidelines:

- The jumbo frames feature is supported on the following:
  - Ethernet ports

Note The following modules only support a maximum of 8092 bytes: WS-X6148-RJ-45V, WS-X6148-RJ21V, WS-X6248-RJ-45, WS-X6248A-RJ-45, WS-X6248-TEL, WS-X6248A-TEL, WS-X6348-RJ-45, WS-X6348-RJ45V, WS-X6348-RJ-21, and WX-X6348-RJ21V.

The WS-X6548-RJ-21 and WS-X6548-RJ-45 modules use different hardware at the PHY level and support the full jumbo frame default value of 9216 bytes.

Note The WS-X6516-GE-TX (10/100/1000) module only supports a maximum of 8092 bytes at the 100 Mbps speed. At 10 Mbps and 1000 Mbps the module supports the jumbo frame default of 9216 bytes.

  - Trunk ports
  - EtherChannels
- Jumbo frames are supported on all Optical Services Modules (OSMs).
- Jumbo frames are not supported on ATM modules (WS-X6101-OC12-SMF/MMF).
- The Multilayer Switching Feature Card 2 (MSFC2) supports routing of jumbo frames.
- The Gigabit Switch Router (GSR) supports routing of jumbo frames.
- The Multilayer Switching Feature Card (MSFC) and Multilayer Switch Module (MSM) do not support jumbo frame routing; if jumbo frames are sent to these routers, router performance is significantly degraded.

Note Occasionally, you might see a *"Jumbo frames inconsistent state"* message for a port or multiple ports after entering the **show port jumbo** command. If this occurs, enter the **set port jumbo** command to reenable the ports.

To enable the jumbo frames feature on an Ethernet port, perform this task in privileged mode:

	Task	Command
Step 1	Enable jumbo frames.	**set port jumbo** *mod/port* **enable**
Step 2	Verify the port configuration.	**show port jumbo**

This example shows how to enable the jumbo frames feature on a port and verify the configuration:

```
Console> (enable) set port jumbo 2/1 enable
Jumbo frames enabled on port 2/1
Console> (enable) show port jumbo
Jumbo frames MTU size is 9216 bytes
Jumbo frames enabled on port(s) 2/1
```

To disable the jumbo frames feature on an Ethernet port, perform this task in privileged mode:

	Task	Command
Step 1	Disable jumbo frames.	**set port jumbo** *mod/port* **disable**
Step 2	Verify the port configuration.	**show port jumbo**

This example shows how to disable the jumbo frames feature on a port:

```
Console> (enable) set port jumbo 2/1 disable
Jumbo frames disabled on port 2/1
Console> (enable)
```

Configuring the Jumbo Frame Feature on MSFC2

With an MSFC2, you can configure the MTU size on VLAN interfaces to support routing of jumbo frames.

The jumbo frame feature supports only a single larger-than-default MTU size on the switch. Configuring a VLAN interface with an MTU size greater than the default automatically configures all other VLAN interfaces that have an MTU size greater than the default to the newly configured size. VLAN interfaces that have not been changed from the default are not affected.

To configure the MTU value, perform this task:

	Task	Command
Step 1	Access VLAN interface configuration mode.	Router(config)# **interface vlan** *vlan_ID*
Step 2	Set the MTU size. Valid values are from 64 to 17952 bytes.[1]	Router(config-if)# **mtu** *mtu_size*
Step 3	Verify the configuration.	Router# **show interface vlan 111**

1. Set the MTU size no larger than 9216, which is the size supported by the supervisor engine.

This example shows how to set the MTU size on a VLAN interface and verify the configuration:

```
Router(config)# interface vlan 111
Router(config-if)# mtu 9216
Router(config-if)# end
Router# show interface vlan 111
<...Output Truncated...>
MTU 9216 bytes, BW 1000000 Kbit, DLY 10 usec,
<...Output Truncated...>
Router#
```

Checking Connectivity

Use the **ping** and **traceroute** commands to test connectivity.

To check connectivity out a port, perform this task in privileged mode:

	Task	Command
Step 1	Ping a remote host that is located out the port you want to test.	**ping** [**-s**] *host* [*packet_size*] [*packet_count*]
Step 2	Trace the hop-by-hop route of packets from the switch to a remote host located out the port you want to test.	**traceroute** *host*
Step 3	If the host is unresponsive, check the IP address and default gateway configured on the switch.	**show interface** **show ip route**

This example shows how to ping a remote host and how to trace the hop-by-hop path of packets through the network using **traceroute**:

```
Console> (enable) ping somehost
somehost is alive
Console> (enable) traceroute somehost
traceroute to somehost.company.com (10.1.2.3), 30 hops max, 40 byte packets
 1 engineering-1.company.com (173.31.192.206) 2 ms 1 ms 1 ms
 2 engineering-2.company.com (173.31.196.204) 2 ms 3 ms 2 ms
 3 gateway_a.company.com (173.16.1.201) 6 ms 3 ms 3 ms
 4 somehost.company.com (10.1.2.3) 3 ms * 2 ms
Console> (enable)
```