

# Chapter 7

## Interface Adjustments

---

# 7

This chapter describes how to make adjustments to—or how to *fine tune*—the interfaces, and how to enable loopback tests. You will find information about the following processes in this chapter:

- Switching and scheduling priorities
- Interface hold queues
- Setting the bandwidth
- Setting delay
- Setting loopback testing on an interface

To adjust or test an interface, you must be in the configuration command collection mode. To enter this mode, type the EXEC command **configure** at the EXEC prompt. Once in the command collection mode, start configuring the interface by entering the **interface** command.

A command summary is included at the end of the chapter.

---

### *Adjusting Serial Interface Characteristics*

The following sections describe adjustments that can be made to serial interfaces.

#### *Specifying Transmit Delay*

Since the MCI and SCI interface cards can send back-to-back data packets over serial interfaces faster than some hosts can receive them, you can specify a minimum dead-time after transmitting a datagram. Do this using the **transmitter-delay** interface subcommand. It is especially useful for serial interfaces. The full syntax of the command follows.

```
transmitter-delay microseconds  
no transmitter-delay
```

The argument *microseconds* specifies the approximate number of microseconds of minimum delay after transmitting a datagram. The default value is zero; the **no transmitter-delay** command restores this default.

**Example:**

This command specifies a delay of 300 microseconds on interface serial 0.

```
interface serial 0
transmitter-delay 300
```

## Configuring DTR Signal Pulsing

The **pulse-time** interface subcommand enables pulsing DTR signals on the Cisco MCI and SCI serial interfaces. The full syntax of the command follows.

**pulse-time** *seconds*  
**no pulse-time**

The argument *seconds* specifies the interval. When the serial line protocol goes down (for example, because of loss of synchronization) the interface hardware is reset and the DTR signal is held inactive for at least the specified interval. This function is useful for handling encrypting or other similar devices that use the toggling of the DTR signal to resynchronize.

The default interval is zero seconds, which is restored by the **no pulse-time** command.

**Example:**

This command enables DTR pulse signals for three seconds on interface serial 2.

```
interface serial 2
pulse-time 3
```

## Configuring for DCE Appliques

You may configure the clock rate on the serial interface of the SCI and MCI cards to an acceptable bit rate using the **clockrate** interface subcommand. The full syntax is as follows:

**clockrate** *speed*  
**no clockrate**

The argument *speed* is the desired clock rate, and may be one of any of the rates in Table 1-1.

**Table 1-1** Legal Bits per Second Values

1200	2400	4800
9600	19200	34800
56000	64000	72000
125000	148000	500000
800000	1000000	1300000
2000000	4000000	

Be aware that the fastest speeds may not work if your cable is too long, and that speeds faster than 148,000 bits per second are too fast for RS-232 signaling. Cisco recommends you only use the synchronous serial RS-232 signal at speeds up to 64,000 bits per second. To permit a faster speed, use an RS-449 or V.35 applique.

Use the **no clockrate** command to remove the clock rate if you change the interface from a DCE to a DTE device.

**Example:**

These sample commands set the clock rate on the first serial interface to 64,000 bits per second.

```
interface serial 0
clockrate 64000
```

---

## *Adjusting Characteristics That Apply to All Interface Types*

The following sections describe adjustments that apply to all interface types.

### *Configuring Switching and Scheduling Priorities*

The normal operation of the network server allows the switching operations to use as much of the central processor as is required. In the event that the network is running unusually heavy loads that do not allow the processor the time to handle the routing protocols, more priority can be given to the system process scheduler using the **scheduler-interval** global configuration command. The full command syntax follows.

```
scheduler-interval milliseconds
no scheduler-interval
```

The **scheduler-interval** command controls the maximum amount of time that may elapse without running the lowest priority system processes. The minimum interval that may be specified is 500 milliseconds; there is no maximum value. The default is to allow high-priority operations to use as much of the central processor as needed. The **no scheduler-interval** command restores that default.

*Example:*

This command changes the low-priority process schedule to an interval of 750 milliseconds.

```
scheduler-interval 750
```

## Controlling Interface Hold Queues

Each network interface in a network server has a hold-queue limit. This limit is the number of data packets that the interface can store in its hold queue before rejecting new packets. When the interface empties the hold queue by one or more packets, the interface can accept new packets again.

To specify the hold-queue limit of an interface, use the **hold-queue** interface subcommand. Its full syntax is as follows:

```
hold-queue length {in | out}  
no hold-queue {in | out}
```

The argument *length* is the maximum number of packets in the queue. The **in** keyword specifies the input queue; the **out** keyword specifies the output queue.

The **no hold-queue** command with the appropriate keyword restores the default values for an interface. There is no fixed upper limit to a queue size.

The input hold queue prevents a single interface from flooding the network server with too many input packets. Further input packets are discarded if the interface has too many input packets outstanding in the system. The default input hold queue is 75 packets. The default output hold-queue limit is 100 packets. This limit prevents a malfunctioning interface from consuming an excessive amount of memory.

If priority output queueing is being used, the length of the four output queues is set using the **priority-list** configuration command. The **hold-queue** command cannot be used to set an output hold queue length in this situation.

For slow links, use a small output hold-queue limit. This approach prevents storing packets at a rate that exceeds the transmission capability of the link. For fast links, use a large output hold-queue limit. A fast link may be busy for a short time (and thus require the hold queue), but can empty the output hold queue quickly when capacity returns.

To display the current hold queue setting and the number of packets discarded because of hold queue overflows, use the EXEC command **show interfaces**.

## Setting Bandwidth

Higher-level protocols may use bandwidth information to make operating decisions. For example, IGRP uses the minimum path bandwidth to determine a routing metric. The TCP protocol adjusts initial retransmission parameters based on the apparent bandwidth of the outgoing interface.

To set a bandwidth value for an interface, use the **bandwidth** interface subcommand. Its full syntax follows:

```
bandwidth kilobits  
no bandwidth
```

The argument *kilobits* specifies the intended bandwidth in kilobits per second. For a full bandwidth DS3, enter the value 45045. Default bandwidth values are set during startup and can be displayed with the EXEC command **show interfaces**.

The **bandwidth** subcommand sets an informational parameter only; you cannot adjust the actual bandwidth of an interface with this subcommand. For some media, such as Ethernet, the bandwidth is fixed; for other media, such as serial lines, you can change the actual bandwidth by adjusting hardware. For both classes of media, you can use the **bandwidth** subcommand to communicate the current bandwidth to the higher-level protocols.

Use the **no bandwidth** command to restore the default values.

## Setting Delay

Higher-level protocols may use delay information to make operating decisions. For example, IGRP can use delay information to differentiate between a satellite link and a land link.

To set a delay value for an interface, use the **delay** interface subcommand. The full syntax of this command follows.

```
delay tens-of-microseconds  
no delay
```

The argument *tens-of-microseconds* specifies the delay for an interface or network segment in tens of microseconds. Default delay values may be displayed with the EXEC command **show interfaces**. The **no delay** command restores the default.

---

**Note:** The **delay** subcommand sets an informational parameter only; you cannot adjust the actual delay of an interface with this subcommand.

---

## Setting Error Count Reset Frequency

The Cisco interface software provides a mechanism for protection against packet overload and resultant recount errors on the MCI interface cards. This mechanism is set using the **error-threshold** interface subcommand, as follows:

**error-threshold** *milliseconds*

The argument *milliseconds* is the frequency at which the error recount will be set. The default value is 1000 milliseconds.

### *Example:*

These commands set the error recount threshold on Ethernet interface 2 to 10,000 milliseconds.

```
interface ethernet 2
error-threshold 10000
```

## Setting and Adjusting Packet Sizes

Each interface has a default maximum packet size or maximum transmission unit (MTU) size. This number generally defaults to the largest size possible for that type interface. For example, the Ethernet MTU size defaults to 1500 bytes. On serial interfaces, the MTU size varies, but cannot be set smaller than 64 bytes. You can adjust the MTU using the **mtu** interface subcommand. Its full syntax is:

**mtu** *bytes*  
**no mtu**

The argument *bytes* is the desired size in bytes. The **no mtu** command restores this value to its original, default value.

---

## Enabling the Loopback Test

The Cisco software provides a loopback test to detect and distinguish equipment malfunctions between line and modem or CSU/DSU (Channel Service Unit/Digital Service Unit) problems on the network server. If correct data transmission is not possible when an interface is in loopback mode, the interface is the source of the problem. The DSU may have similar loopback functions you can use to isolate the problem, if the interface loopback test passes. If the device does not support local loopback, then this function will have no effect.

You can specify hardware loopback tests on the Cisco Ethernet and serial interfaces, and all Token Ring interfaces except the CSC-R 4 megabit card that are attached to CSU/DSUs, and that support the local loopback signal. The CSU/DSU acts as a DCE device; the Cisco terminal server as a DTE device. The local loopback test generates a CSU loop—a signal that goes through the CSU/DSU to the line, then back through the CSU/DSU to the Cisco terminal server.

The **ping** command can also be useful during loopback operation; it is described in the section on “Testing Connectivity with the Ping Command” in Chapter 5.

The following sections describe the various loopback tests available for the supported interfaces.

### *Enabling Loopback on MCI and SCI Serial Cards*

The MCI and SCI serial interface cards support the loopback function when a CSU/DSU or equivalent device is attached to the Cisco terminal server. Use the **loopback** command to enable loopback on the interface. The full syntax of this command follows:

```
loopback  
no loopback
```

The **loopback** interface subcommand loops the packets through the CSU/DSU to configure a CSU loop, when the device supports this feature. The **no loopback** command disables the function.

### *Enabling Loopback on MCI and MEC Ethernet Cards*

Cisco-designed Ethernet interfaces may also be placed into loopback mode. During loopback operation, the interface receives back every packet it sends. Loopback operation has the additional effect of disconnecting network server functionality from the network. Use these commands to enable or disable the loopback test:

```
loopback  
no loopback
```

### *Enabling Loopback on Token Ring Cards*

All of Cisco’s Token Ring interface cards (except the four megabit CSC-R card) may also be placed into loopback mode. During loopback operation, the interface receives back every packet it sends. Loopback operation has the additional effect of “disconnecting” network server functionality from the network. Use these commands to enable or disable loopback on the interface:

```
loopback  
no loopback
```

## Configuring the Ethernet Loopback Server

The Cisco network server provides an Ethernet loopback server that supports Xerox, Intel, and DEC systems specified by the “blue book,” a joint specification written by Xerox, Digital Equipment Corporation, and Intel that defines the Ethernet protocol. The loopback server responds to forward data loopback messages sent either to the server’s MAC address, or to the broadcast address. Currently, the Ethernet loopback server does not respond to the loopback assistance multicast address.

Use the Ethernet loopback server to test communications between Cisco internetworking products and DEC systems that do not support the IP **ping** command, such as DECnet-only VMS systems.

To originate a loop test on your VMS system with a Cisco server, use the DEC Network Control Program (NCP) command **Loop Circuit**. For more information about the **Loop Circuit** command, consult the DECnet VAX documentation. Cisco network servers support all options that can be specified by the VMS hosts.

---

## Interface Configuration Subcommand Summary

This section provides an alphabetical list of all the interface commands described in this chapter.

### **[no] bandwidth** *kilobits*

Sets a bandwidth value for an interface. The argument *kilobits* specifies the intended bandwidth in kilobits per second. Default bandwidth values are set during startup and can be displayed with the EXEC command **show interfaces**. The **no** form of the command restores the default.

### **[no] clockrate** *speed*

Configures the clock rate on the serial interface of the SCI and MCI cards to an acceptable bit rate. The argument *speed* is the desired clock rate in bits per second. The **no** form removes the command from the configuration.

### **[no] delay** *tens-of-microseconds*

Sets a delay value for an interface. The argument *tens-of-microseconds* specifies the delay for an interface or network segment in tens of microseconds. Default delay values may be displayed with the EXEC command **show interfaces**. The **no** form of the command restores the default. The **delay** subcommand sets an informational parameter only; you cannot adjust the actual delay of an interface with this subcommand.

**[no] hold-queue** *length* {**in** | **out**}

Specifies the hold-queue limit of an interface. The argument *length* is the maximum number of packets in the queue. The **in** keyword specifies the input queue; the **out** keyword specifies the output queue. Use the **no hold-queue** command to restore the default values for an interface. There is no fixed upper limit to a queue size.

**[no] loopback**

On MCI and SCI serial cards—Loops the packets through the CSU/DSU to configure a “CSU loop,” when the device supports this feature. This is similar to the **loopback line** command on the HSSI.

On MCI and MEC Ethernet cards—Loops the packets at the interface within the terminal server.

On the CSC-R16 card—Loops the packets at the interface within the terminal server. The **no** form of the command disables the loopback test.

**[no] pulse-time** *seconds*

Enables pulsing DTR signals on the Cisco MCI and SCI serial interfaces for a minimum interval of *seconds*.

**[no] scheduler-interval** *milliseconds*

Controls the maximum amount of time that may elapse without running the lowest priority system processes. The minimum interval that may be specified is 500 milliseconds; there is no maximum value. The default is to allow high priority operations to use as much of the central processor as needed. The **no scheduler-interval** command restores that default.

**[no] transmitter-delay** *microseconds*

Specifies a minimum dead-time after transmitting a datagram. The argument *microseconds* specifies the approximate number of microseconds of minimum delay after transmitting a datagram. For MCI and SCI serial cards only. The **no** form of the command restores the default value of zero microseconds.

