CHAPTER 6

Interface Configuration Commands

This chapter contains the commands used to configure nonprotocol-specific interface features. The commands are in alphabetical order. For hardware technical descriptions, and for information about installing the communication server interfaces, refer to the hardware installation and maintenance publication for your particular product.

For interface configuration tasks and examples, refer to the *Communication Server Configuration Guide*.

atm-dxi map

To map a given VPI and VCI to a Frame Relay DLCI, use the **atm-dxt map** interface configuration command. Use the **no** form of this command to remove the definition.

atm-dxi map *protocol address VPI VCI* [**broadcast**]
no atm-dxi map *protocol address*

Syntax Description

protocol	Specifies the protocol: **ip**, **novell**
address	Protocol-specific address.
VPI	Specifies the Virtual Path Identifier in the range 1 to 15.
VCI	Specifies the Virtual Circuit Identifier in the range 1 to 63.
broadcast	(Optional.) Broadcasts should be forwarded to this address.

Default

None

Command Mode

Interface configuration

Usage Guidelines

This command is used in configurations where the communication server is intended to communicate with an ATM network through an ATM Data Service Unit (ADSU). Given the circuit identifier parameters (VPI/VCI) for the ATM virtual circuit, the DXI Frame Address (DFA) that is used for communication between the communication server and the ADSU is computed and used.

Example

In the following example, all IP packets intended for the host with IP address 131.108.170.49 are converted into ATM cells identified with a VPI of 2 (binary 0000 0010) and a VCI of 46 (binary 0000 0000 0010 1110) by the ADSU. Using the mapping defined in Annex A of the ATM DXI Specification, this combination of VPI and VCI maps into to a DFA of 558 (binary 1000101110). The ADSU will then extract the VPI and VCI information from the DFA of the incoming frame relay frame when formulating ATM cells.

```
interface serial 0
atm-dxi map ip 131.108.170.49 2 46 broadcast
```

Related Command

encapsulation atm-dxi

backup delay

To define how much time should elapse before a secondary line is set up or taken down after a primary line transition, use the **backup delay** interface configuration command. Use the **no backup delay** command to remove the definition.

backup delay {*enable-delay* | **never**} {*disable-delay* | **never**}
no backup delay {*enable-delay* | **never**} {*disable-delay* | **never**}

Syntax Description

enable-delay	Integer argument that specifies the delay in seconds after the primary line goes down before the secondary line is activated.
disable-delay	Integer argument that specifies the delay in seconds after the primary line goes up before the secondary line is deactivated.
never	Keyword that is specified to prevent the secondary line from being activated or deactivated.

Default

never

Command Mode

Interface configuration.

Usage Guidelines

When a primary line goes down, the communication server delays the amount of seconds defined by the *enable-delay* argument before enabling the secondary line. If, after the delay period, the primary line is still down, the secondary line is activated.

When a primary line comes back up, the communication server will delay the amount of seconds defined by the *disable-delay* argument.

In cases where there are spurious signal disruptions that may appear as intermittent lost carrier signals, it is recommended that some delay be enabled before activating and deactivating a secondary.

The interval configured with the **backup delay** command does not affect the operation of the **backup load** command.

Example

The following example sets a 10-second delay on deactivating the secondary line; however, the line is activated immediately:

```
interface serial 0
backup delay 0 10
```

backup interface

To configure the serial interface as a secondary, or dial backup line, use the **backup interface** interface configuration command. Use the **no backup** command with the appropriate serial port designation to turn this feature off.

backup interface *interface-name*
no backup interface *interface-name*

Syntax Description

interface-name or type Serial port to be set as the secondary interface line.

Default

None

Command Mode

Interface configuration

Example

The following example sets serial 1 as the backup line to serial 0:

```
interface serial 0
backup interface serial 1
```

Related Command

down-when-looped

backup load

To set the traffic load thresholds for dial backup service, use the **backup load** interface configuration command. Use the **no backup load** command to remove the setting.

backup load {*enable-threshold* | **never**} {*disable-load* | **never**}
no backup load {*enable-threshold* | **never**} {*disable-load* | **never**}

Syntax Description

enable-threshold	Integer argument that specifies a percentage of the primary line's available bandwidth.
disable-load	Integer argument that specifies a percentage of the primary line's available bandwidth.
never	Keyword that sets the secondary line to never be activated due to load.

Default

never

Command Mode

Interface configuration

Usage Guidelines

When the transmitted or received load on the primary line is greater than the value assigned to the *enable-threshold* argument, the secondary line is enabled.

When the transmitted load on the primary line plus the transmitted load on the secondary line is less than the value entered for the *disable-load* argument, and the received load on the primary line plus the received load on the secondary line is less than the value entered for the *disable-load* argument, the secondary line is disabled.

If the **never** keyword is used instead of an *enable-threshold* value, the secondary line is never activated because of load. If the **never** keyword is used instead of an *disable-load* value, the secondary line is never deactivated because of load.

Example

The following example sets the traffic load threshold to 60 percent on the primary line. When that load is exceeded, the secondary line is activated, and will not be deactivated until the combined load is less than 5 percent of the primary bandwidth.

```
interface serial 0
backup load 60 5
```

bandwidth

To set a bandwidth value for an interface, use the **bandwidth** interface configuration command. Use the **no bandwidth** command to restore the default values.

bandwidth *kilobits*
no bandwidth

Syntax Description

kilobits	Intended bandwidth in kilobits per second. For a full bandwidth DS3, enter the value 44736.

Default

Default bandwidth values are set during startup.

Command Mode

Interface configuration

Usage Guidelines

Bandwidth values can be displayed with the EXEC command **show interfaces**.

The **bandwidth** command sets an informational parameter only; you cannot adjust the actual bandwidth of an interface with this command. 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** configuration command to communicate the current bandwidth to the higher-level protocols.

Additionally, 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.

At higher bandwidths, the value you configure with the **bandwidth** command is not what is displayed by the **show interface** command. The value shown is that used in IGRP updates and also used in computing load.

Note This is a routing parameter only; it does not affect the physical interface.

Example

The following example sets the full bandwidth for DS3 transmissions:

```
interface serial 0
bandwidth 44736
```

clear counters

To clear the interface counters, use the **clear counters** EXEC command.

clear counters [*type number*]

Syntax Description

type (Optional.) Specifies the interface type; it is one of the keywords listed in Table 1-1.

number (Optional.) Specifies the interface counter displayed with the **show interfaces** command.

Table 1-1 Clear Counters Interface Type Keywords

Keyword	Interface Type
async	Async interface
dialer	Dialer interface
ethernet	Ethernet interface
loopback	Loopback interface
null	Null interface
serial	Synchronous serial interface
tokenring	Token Ring interface
tunnel	Tunnel interface

Command Mode

EXEC

Usage Guidelines

This command clears all the current interface counters from the interface unless the optional arguments *type-keyword* and *number* are specified to clear only a specific interface type (serial, Ethernet, Token Ring, and so on).

Note This command will not clear counters retrieved using SNMP, but only those seen with the EXEC **show interface** command.

Example

The following example illustrates how to clear all interface counters:

```
cs# clear counters
```

Related Command

show interfaces

clear interface

To reset the hardware logic on an interface, use the **clear interface** EXEC command.

clear interface *type number*

Syntax Description

type Specifies the interface type; it is one of the keywords listed in Table 1-2.

number Specifies the port, connector, or interface card number.

Table 1-2 Clear Interface Type Keywords

Keyword	Interface Type
async	Async interface
ethernet	Ethernet interface
loopback	Loopback interface
null	Null interface
serial	Synchronous serial interface
tokenring	Token Ring interface
tunnel	Tunnel interface

Command Mode

EXEC

Note Under normal circumstances, you do not need to clear the hardware logic on interfaces.

Example

```
CS# clear interface hssi 1
```

clear rif-cache

To clear entries from the Routing Information Field (RIF) cache, use the **clear rif-cache** EXEC command.

clear rif-cache

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Example

The following example illustrates how to clear the RIF cache:

```
cs# clear rif-cache
```

Related Commands

multiring

clockrate

To configure the clock rate for appliques (connector hardware) on the serial interface of the MCI and SCI cards to an acceptable bit rate, use the **clockrate** interface configuration command. Use the **no clockrate** command to remove the clock rate if you change the interface from a DCE to a DTE device.

clockrate *bps*
no clockrate

Syntax Description

bps	Desired clock rate in bits per second: 1200, 2400, 4800, 9600, 19200, 34800, 56000, 64000, 72000, 125000, 148000, 500000, 800000, 1000000, 1300000, 2000000, or 4000000.

Default

No clock rate

Command Mode

Interface configuration

Usage Guidelines

Be aware that the fastest speeds might 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. It is recommended that 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.

Example

The following example sets the clock rate on the first serial interface to 64,000 bits per second:

```
interface serial 0
clockrate 64000
```

custom-queue-list

To assign a custom queue list to an interface, use the **custom-queue-list** interface configuration command. To remove a specific list or all list assignments, use the **no** form of the command.

custom-queue-list *list*
no custom-queue-list [*list*]

Syntax Description

list	Number of the custom queue list you want to assign to the interface. An integer from 1 to 10.

Default

None

Command Mode

Interface configuration

Usage Guidelines

Only one queue list can be assigned per interface. Use this command in place of the **priority-list** command (not in addition to it). Custom queuing allows a fairness not provided with priority queuing. With custom queueing, you can control the interfaces' available bandwidth when it is unable to accommodate the aggregate traffic enqueued. Associated with each output queue is a configurable byte count, which specifies how many bytes of data should be delivered from the current queue by the system before the system moves on to the next queue. When a particular queue is being processed, packets are sent until the number of bytes sent exceeds the queue byte count or until the queue is empty.

Example

In the following example, custom queue list number 3 is assigned to interface serial 0:

```
cs(config)# interface serial 0
cs(config-if)# custom-queue-list 3
```

delay

To set a delay value for an interface, use the **delay** interface configuration command. Use the **no delay** command to restore the default delay value.

delay *tens-of-microseconds*
no delay

Syntax Description

tens-of-microseconds	Integer that specifies the delay in tens of microseconds for an interface or network segment.

Default

Default delay values may be displayed with the EXEC command **show interfaces**.

Command Mode

Interface configuration

Example

The following example sets a 30,000-microsecond delay on interface serial 3:

```
interface serial 3
delay 30000
```

Related Command

show interfaces

description

To add a description to an interface configuration, use the **description** interface configuration command. Use the **no description** command to remove the description.

description *string*
no description

Syntax Description

string	Comment or a description to help you remember what is attached to this interface.

Default

None

Command Mode

Interface configuration

Usage Guidelines

The **description** command is meant solely as a comment to be put in the configuration to help you remember what certain interfaces are used for. The description appears in the output of the following EXEC commands: **show configuration, show interfaces**, and **write terminal.**

Example

The following example describes a 3174 controller on interface serial 0:

```
interface serial 0
description 3174 Controller for test lab
```

Related Commands

A dagger (†) indicates that the command is documented in another chapter.

show configuration †
show interfaces
write terminal †

down-when-looped

To configure an interface to inform the system it is down when loopback is detected, use the **down-when-looped** interface configuration command.

down-when-looped

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Interface configuration

Usage Guidelines

This command is valid for PPP encapsulation on serial and HSSI interfaces.

When an interface has a backup interface configured, it is often desirable that the backup interface be enabled when the primary interface is either down or in loopback. By default, the backup is only enabled if the primary interface is down. By using the **down-when-looped** command, the backup interface will also be enabled if the primary interface is in loopback.

If testing an interface with the loopback command, or by placing the DCE into loopback, **down-when-looped** should not be configured; otherwise packets will not be transmitted out the interface that is being tested.

Example

In the following example, interface serial 0 is configured for PPP encapsulation. It is then configured to let the system know that it is down when in loopback mode.

```
cs(config)# interface serial0
cs(config-if)# encapsulation ppp
cs(config-if)# down-when-looped
cs(config-if)
```

Related Commands

backup interface
loopback

early-token-release

To enable early token release, use the **early-token-release** interface configuration command. This feature helps to increase the total bandwidth of the Token Ring.

early-token-release
no early-token-release

Syntax Description

This command has no arguments or keywords.

Default

By default, early token release is not enabled on the interface.

Command Mode

Interface configuration

Usage Guidelines

Early token release is a method whereby the Token Ring interfaces can release the token back onto the ring immediately after transmitting rather than waiting for the frame to return.

The CSC-R16M, CSC-2R, and CSC-1R cards support early token release. Once enabled, use the **no early-token-release** command to disable this feature.

Example

The following example enables the use of early token release on interface Token Ring 1:

```
interface tokenring 1
early-token-release
```

encapsulation

To set the encapsulation method used by the interface, use the standard Ethernet Version 2.0 encapsulation, use the **encapsulation** interface configuration command.

encapsulation *encapsulation-type*

Syntax Description

encapsulation-type Encapsulation type. See Table 1-3 for a list of supported encapsulation types.

Table 1-3 Encapsulation Types

Keyword	Encapsulation Type
atm-dxi	Asynchronous Transfer Mode-Data Exchange Interface.
bfex25	Blacker Front End Encryption X.25 operation (for serial interface)
ddnx25-dce	DDN X.25 DCE operation (for serial interface)
ddnx25	DDN X.25 DTE operation (for serial interface)
frame-relay	Frame Relay (for serial interface)
lapb-dce	X.25 LAPB DCE operation (for serial interface)
lapb	X.25 LAPB DTE operation (for serial interface)
multi-lapb-dce	X.25 LAPB multiprotocol DCE operation (for serial interface)
multip-lapb	X.25 LAPB multiprotocol DTE operation (for serial interface)
ppp	Point-to-Point Protocol (PPP) (for serial interface)
smds	Switched Multimegabit Data Services (SMDS) (for serial interface)
X25-dce	X.25 DCE operation (for serial interface)
x25	X.25 DTE operation (for serial interface)

Default

The default depends on the type of interface. For example, an Ethernet interface defaults to ARPA.

Command Mode

Interface configuration

Usage Guidelines

To change the encapsulation on LAN media, do not use the **encapsulation** command. Instead, change the encapsulation on a per-protocol basis.

Examples

The following example reenables standard Ethernet Version 2.0 encapsulation on interface Ethernet 0:

```
interface ethernet 0
encapsulation arpa
```

The following example sets IEEE 802.3 encapsulation on Ethernet 1:

```
interface ethernet 1
encapsulation sap
```

The following example enables PPP encapsulation on interface serial 0:

```
interface serial 0
encapsulation ppp
```

The following example sets IEEE 802.2 encapsulation on Ethernet 1:

```
interface ethernet 1
encapsulation snap
```

Related Commands

A dagger (†) indicates that the command is documented in another chapter.

keepalive
ppp †
ppp authentication chap
slip †

error-threshold

To set the mechanism that protects against packet overload and resulting recount errors on the MCI interface cards, use the **error-threshold** interface configuration command.

error-threshold *milliseconds*

Syntax Description

milliseconds	Frequency at which the error recount will be set in milliseconds.

Default

1000 milliseconds

Command Mode

Interface configuration

Example

The following commands set the error recount threshold on Ethernet interface 2 to 10,000 milliseconds:

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

hold-queue

To specify the hold-queue limit of an interface, use the **hold-queue** interface configuration command. Use the **no hold-queue** command with the appropriate keyword to restore the default values for an interface.

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

Syntax Description

length	An integer that specifies the maximum number of packets in the queue.
in	A keyword that specifies the input queue.
out	A keyword that specifies the output queue.

Default

The default input hold-queue limit is 75 packets. The default output hold-queue limit is 40 packets. These limits prevent a malfunctioning interface from consuming an excessive amount of memory. There is no fixed upper limit to a queue size.

Command Mode

Interface configuration

Usage Guidelines

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.

If priority output queueing is being used, the length of the four output queues is set using the **priority-list** global 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**.

Note Increasing the hold queue can have detrimental effects on network routing and response times. For protocols that use seq/ack packets to determine round trip times, do not increase the output queue. Dropping packets instead informs hosts to slow down transmissions to match available bandwidth. This is generally better than having duplicate copies of the same packet within the network (which can happen with large hold queues).

Example

The following example illustrates how to set a small input queue on a slow serial line:

```
interface serial 0
hold-queue 30 in
```

Related Command

show interfaces

interface

To configure an interface type and enter interface configuration mode, use the **interface** global configuration command.

interface *interface-type interface-number*

To configure a subinterface, use the **interface** global configuration command.

interface *interface-type interface-number.subinterface-number* [**multipoint** | **point-to-point**]

Syntax Description

interface-type	Specifies the type of interface to be configured. See Table 1-4.
interface-number	Specifies the port, connector, or interface card number. The numbers are assigned at the factory at the time of installation or when added to a system, and can be displayed with the **show interfaces** command.
subinterface-number	Specifies the subinterface number in the range 1 to 4294967293. The *interface-number* that precedes the period (.) must match the *interface-number* this subinterface belongs to.
multipoint | **point-to-point**	Specifies a multipoint or point-to-point subinterface. Default is **multipoint**.

Table 1-4 Interface Type Keywords

Keyword	Interface Type
async	Line used as an asynchronous interface.
dialer	Dialer interface.
ethernet	Ethernet IEEE 802.3 interface.
loopback	Loopback interface. The interface-number is the number of the loopback interface that you want to create or configure.There is no limit on the number of loopback interfaces you can create.
null	Null interface.
serial	Serial interface.
tokenring	Token Ring interface.
tunnel	Tunnel interface. The *interface-number* is the number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.

Default

The default mode for subinterfaces is **multipoint**.

Command Mode

Global configuration

Usage Guidelines

Subinterfaces can be configured to support partially-meshed Frame Relay networks and multiple IPX encapsulations on LAN media (refer to the *Communication Server Configuration Guide*).

Examples

In the following example, serial interface 0 is configured with PPP encapsulation:

```
interface serial 0
encapsulation ppp
```

The following example enables loopback mode and assigns an IP network address and network mask to the interface. The loopback interface established here will always appear to be up:

```
interface loopback 0
ip address 131.108.1.1 255.255.255.0
```

The following example shows how a partially-meshed frame relay network can be configured. In this example, subinterface serial 0.1 is configured as a multipoint subinterface with three frame relay PVCs associated, and subinterface serial 0.2 is configured as a point-to-point subinterface.

```
interface serial 0
encapsulation frame-relay
interface serial 0.1 multipoint
ip address 131.108.10.1 255.255.255.0
frame-relay interface-dlci 42 broadcast
frame-relay interface-dlci 53 broadcast
interface serial 0.2 point-to-point
ip address 131.108.11.1 255.255.0
frame-relay interface-dlci 59 broadcast
```

Related Commands

show interfaces

interface dialer

To designate a dialer rotary group leader, use the **interface dialer** interface configuration command.

interface dialer *interface-number*

Syntax Description

number	Integer that you select to indicate a dialer rotary group in the range 0 to 9.

Default

None

Command Mode

Interface configuration

Usage Guidelines

Dialer rotary groups allow you to apply a single interface configuration to a set of interfaces. Once the interface configuration is propagated to a set of interfaces, those interfaces can be used to place calls using the standard dial-on-demand criteria. When many destinations are configured, any of these interfaces can be used for outgoing calls.

Dialer rotary groups are useful in environments that require many calling destinations. Only the rotary group needs to be configured with all of the **dialer map** commands. The only configuration required for the interfaces is the **dialer rotary-group** command indicating that each interface is part of a dialer rotary group.

Although a dialer rotary group is configured as an interface, it is not a physical interface. Instead it represents a group of interfaces. Any number of dialer groups can be defined.

Interface configuration commands entered after the **interface dialer** command will be applied to all physical interfaces assigned to specified rotary group.

Example

The following example identifies interface dialer 1 as the dialer rotary group leader. Interface dialer 1 is not a physical interface, but represents a group of interfaces. The interface configuration commands that follow apply to all interfaces included in this group.

```
interface dialer 1
encapsulation ppp
dialer in-band
dialer map ip 131.108.2.5 username YYY 14155553434
dialer map ip 131.126.4.5 username ZZZ
```

Related Commands

A dagger (†) indicates that the command is documented in another chapter.

dialer rotary-group †

loopback

To diagnose equipment malfunctions between an interface and a device, use the **loopback** interface configuration command. The **no loopback** command disables the test.

loopback
no loopback

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Interface configuration

Usage Guidelines

On HSSI serial interface cards, the loopback function configures a two-way internal and external loop on the HSA applique of the specific interface.

On MCI and SCI serial interface cards, the loopback functions when a CSU/DSU or equivalent device is attached to the communication server. The **loopback** command loops the packets through the CSU/DSU to configure a CSU loop, when the device supports this feature.

On the MCI and MEC Ethernet cards, the interface receives back every packet it sends when the **loopback** command is enabled. Loopback operation has the additional effect of disconnecting network server functionality from the network.

On all Token Ring interface cards (except the 4-megabit CSC-R card), the interface receives back every packet it sends when the **loopback** command is enabled. Loopback operation has the additional effect of disconnecting network server functionality from the network.

Example

The following example configures the loopback test on interface Ethernet 4:

```
interface ethernet 4
loopback
```

Related Command

down-when-looped

loopback applique

To loop internally on the HSSI applique, testing communication, use the **loopback** interface configuration command. To remove the loop, use the **no** form of the command.

loopback applique
no loopback applique

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Interface configuration

Usage Guidelines

This command loops the packets within the applique, thus providing a way to test for communication within the communication server.

Example

The following example configures the loopback test on the HSSI applique:

```
interface serial 1
loopback applique
```

loopback dte

To loop packets to DTE internally within the CSU/DSU at the DTE interface, use the **loopback** interface configuration command. To remove the loop, use the **no** form of the command.

loopback dte
no loopback dte

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Interface configuration

Example

The following example configures the loopback test on the DTE interface:

```
interface serial 1
loopback dte
```

loopback line

To loop packets completely through the CSU/DSU to configure the CSU loop, use the **loopback line** interface configuration command. To remove the loop, use the **no** form of the command.

loopback line
no loopback line

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Interface configuration

Usage Guidelines

This command is useful for testing the DCE device (CSU/DSU) itself.

Example

The following example configures the loopback test on the DCE device:

```
interface serial 1
loopback line
```

loopback remote

To loop packets completely through the CSU/DSU, over the DS3 link, to the remote CSU/DSU and back, use the **loopback remote** interface configuration command. To remove the loop, use the **no** form of the command.

loopback remote
no loopback remote

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Interface configuration

Usage Guidelines

This command is useful for testing the DCE device (CSU/DSU) itself.

Example

The following example configures a remote loopback test:

```
interface serial 0
loopback remote
```

mop enabled

To enable an interface to support the Maintenance Operation Protocol (MOP), use the **mop enabled** interface configuration command. To disable MOP on an interface, use the **no mop enabled** command.

mop enabled
no mop enabled

Syntax Description

This command has no arguments or keywords.

Default

Enabled by default on Ethernet interfaces and disabled on all other interfaces.

Command Mode

Interface configuration

Example

In the following example, MOP is enabled for serial interface 0:

```
cs(config)# interface serial0
cs(config-if)# mop enabled
```

Related Commands

A dagger (†) indicates that the command is documented in another chapter.

mop sysid
mop restransmit-timer †
mop retries †

mop sysid

To enable an interface to send out period Maintenance Operation Protocol (MOP) system identification messages, use the **mop sysid** interface configuration command. To disable MOP message support on an interface, use the **no mop sysid** command.

mop sysid
no mop sysid

Syntax Description

This command has no arguments or keywords.

Default

Enabled

Command Mode

Interface configuration

Usage Guidelines

you can still run MOP without having the background system ID messages sent. This lets you use the MOP remote console, but does not generate messages used by the configurator.

Example

In the following example, serial interface 0 is enabled to send MOP system identification messages:

```
communication server(config)# interface serial0
communication server(config-if)# mop sysid
```

Related Commands

A dagger (†) indicates that the command is documented in another chapter.

mop device-code †
mop enabled

mtu

To adjust the maximum packet size or maximum transmission unit (MTU) size, use the **mtu** interface configuration command. Use the **no mtu** command to restore the MTU value to its original default value.

mtu *bytes*
no mtu

Syntax Description

bytes	Desired size in bytes.

Default

Table 1-5 lists default MTU values according to media type.

Table 1-5 Default Media MTU Values

Media Type	Default MTU
Ethernet	1500
Serial	1500
Token Ring	4464

Command Mode

Interface configuration

Usage Guidelines

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. On serial interfaces, the MTU size varies, but cannot be set smaller than 64 bytes.

Note Changing the MTU value with the **mtu** interface configuration command can affect values for the protocol-specific versions of the command (**ip mtu** for example). If the values specified with the **ip mtu** interface configuration command is the same as the value specified with the **mtu** interface configuration command, and you change the value for the **mtu** interface configuration command, the **ip mtu** value automatically matches the new **mtu** interface configuration command value. However, changing the values for the **ip mtu** configuration commands has no effect on the value for the **mtu** interface configuration command.

Example

The following example specifies an MTU of 1000 bytes:

```
interface serial 1
mtu 1000
```

Related Commands

A dagger (†) indicates that the command is documented in another chapter.

encapsulation smds †
ip mtu †

multiring ip

To enable collection and use of RIF information, use the **multiring ip** interface subcommand. The **no multiring ip** subcommand disables the use of RIF information for the protocol specified.

multiring ip
no multiring ip

Syntax Description

These commands have no arguments or keywords.

Default

None

Type

Interface subcommand

Usage Guidelines

ASM-CS communication servers on a Token Ring network in a source-route bridging environment must support the collection and use of RIF information. This must be done to provide necessary path information to the host. When the **multiring ip** command is enabled, the router will source packets that include information used by source-route bridges. This allows the ASM-CS with Token Ring interfaces to connect to a source-route bridged Token Ring network.

Level 3 routers that use protocol-specific information (for example, Novell IPX or XNS headers) rather than MAC information to route datagrams also must be able to collect and use RIF information to ensure that the Level 3 routers can transmit datagrams across a source-route bridge. The software default is to not collect and use RIF information for routed protocols. This allows operation with software that does not understand or properly use RIF information, such as versions of Novell NetWare prior to Version 2.15c.

Example

The following example enables a Token Ring interface. RIFs will be generated for IP frames.

```
interface tokenring 0
multiring ip
ip address 131.108.183.37  255.255.255.0
```

Related Commands

rif
rif timeout

ntp disable

To prevent an interface from receiving NTP packets, use the **ntp disable** interface configuration command. To enable receipt of NTP packets on an interface, use the **no ntp disable** command.

ntp disable
no ntp disable

Syntax Description

This command has no arguments or keywords.

Default

Enabled

Command Mode

Interface configuration

Usage Guidelines

This command provides a simple method of access control.

Example

In the following example, interface Ethernet 0 is prevented from receiving NTP packets:

```
cs(config)# interface ethernet0
cs(config-if)# ntp disable
```

ppp authentication chap

To enable Challenge Handshake Authentication Protocol (CHAP) on a serial interface, use the **ppp authentication chap** interface configuration command. Use the **no ppp authentication chap** command to disable this encapsulation.

ppp authentication chap
no ppp authentication chap

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Interface configuration

Usage Guidelines

Once you have enabled CHAP, the local communication server requires a password from remote devices. If the remote device does not support CHAP, no traffic will be passed to that device.

Example

The following example enables CHAP on interface serial 4:

```
interface serial 4
encapsulation ppp
ppp authentication chap
```

Related Command

encapsulation ppp

ppp authentication pap

To enable Password Authentication Protocol (PAP) on a serial interface, use the **ppp authenticate pap** interface configuration command. To disable this feature, use the **no** form of this command.

ppp authentication pap
no ppp authentication pap

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Interface configuration

Usage Guidelines

When you enable PAP, it forces the router to negotiate with the remote device for the Password Authentication Protocol. If the remote device does not support PAP, no traffic will be passed to that device.

If both sides of the serial link are running PPP and PAP, both username entries and their passwords must be present on both sides of the link. You define these with the **username** *name* **password** *secret* global configuration command.

Example

The following example enables PAP on serial interface 4:

```
interface serial 4
ppp authentication pap
```

Related Commands

encapsulation ppp
username password

ppp quality

To enable Link Quality Monitoring (LQM) on a serial interface, use the **ppp quality** interface configuration command. Use the **no** form of this command to disable LQM.

ppp quality *percentage*
no ppp quality

Syntax Description

percentage	Specifies the link quality threshold. The range is 1 to 100.

Default

Disabled

Command Mode

Interface configuration

Usage Guidelines

The percentages are calculated for both incoming and outgoing directions. The outgoing quality is calculated by comparing the total number of packets and bytes sent to the total number of packets and bytes received by the peer. The incoming quality is calculated by comparing the total number of packets and bytes received to the total number of packets and bytes sent by the peer.

If the link quality percentage is not maintained, the link is deemed to be of poor quality and is taken down.The policy implements a time lag so that the link does not bounce up and down.

Example

The following example enables LQM on interface serial 4:

```
interface serial 4
encapsulation ppp
ppp quality 80
```

Related Commands

A dagger (†) indicates that the command is documented in another chapter.

encapsulation ppp
keepalive †

priority-group

To assign the specified priority list to an interface, use the **priority-group** interface configuration command. Use the **no priority-group** command to remove the specified **priority-group** assignment.

priority-group *list*
no priority-group

Syntax Description

list Priority list number assigned to the interface.

Default

None

Command Mode

Interface configuration

Usage Guidelines

Only one list can be assigned per interface. Priority output queueing provides a mechanism to prioritize packets transmitted on an interface.

Example

The following example causes packets on interface serial 0 to be classified by priority list 1:

```
interface serial 0
priority-group 1
```

Related Commands

A dagger (†) indicates that the command is documented in another chapter.

priority-list †
priority-list interface †
priority-list queue-limit †

pulse-time

To enable pulsing DTR signal intervals on the serial interfaces, use the **pulse-time** interface configuration command. Use the **no pulse-time** command to restore the default interval.

pulse-time *seconds*
no pulse-time

Syntax Description

seconds	Integer that specifies the DTR signal interval in seconds.

Default

0 seconds

Command Mode

Interface configuration

Usage Guidelines

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.

Example

The following example enables DTR pulse signals for three seconds on interface serial:

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

rif

To enter static source-route information into the RIF cache, use the **rif** global configuration command. The command **no rif** *MAC-address* removes an entry from the cache.

rif *MAC-address* [*RIF-string*] [*interface-name*]
no rif *MAC-address* [*interface-name*]

Syntax Description

MAC-address	A 12-digit hexadecimal string written as a dotted triple; for example 0010.0a00.20a6
RIF-string	Optional argument used to specify a series of 4-digit hexadecimal numbers separated by a dot (.). This RIF string is inserted into the packets sent to the specified MAC address
interface-name	Optional argument used to specify an interface name (for example, tokenring0) that indicates the origin of the RIF

Default

None

Type

Global

Usage Guidelines

ASM-CS communication servers on a Token Ring network in a source-route bridging environment must support the collection and use of RIF information. This must be done to provide necessary path information to the host.

If a Token Ring host does not support the use of IEEE 802.2 TEST or XID datagrams as explorer packets, you may need to add static information to the RIF cache.

Using the **rif** command without any of the optional arguments puts an entry into the RIF cache indicating that packets for this MAC address should not have RIF information.

Do not configure a static RIF with any of the *all rings* type codes. Doing so causes traffic for the configured host to appear on more than one ring and leads to unnecessary congestion.

Example

The following example illustrates a RIF that describes a two-hop path as 0830.0155.100a.5550.

```
rif 1000.5A01.0203 0830.0155.100a.5550
```

Related Command

multiring ip
rif timeout

rif timeout

To determine the number of minutes an inactive RIF entry is kept, use the global configuration command **rif timeout**. The **no rif timeout** command restores the default.

rif timeout *minutes*
no rif timeout

Syntax Description

minutes	Assigns a new interval value using the *minutes* argument. The minimum value is one minute.

Default

15 minutes

Type

Global

Usage Guidelines

ASM-CS communication servers on a Token Ring network in a source-route bridging environment must support the collection and use of RIF information. This must be done to provide necessary path information to the host.

RIF information is maintained in a cache whose entries are aged. The EXEC command **show rif** displays the contents of the RIF cache. The EXEC command **clear rif-cache** clears the contents of the RIF cache.

Example

The following example changes the timeout period to five minutes.

```
rif timeout 5
```

Related Commands

clear rif-cache
multiring ip
rif
show rif

ring-speed

To set the ring speed for the CSC-1R and CSC-2R Token Ring interfaces, use the **ring-speed** interface configuration command.

ring-speed *speed*

Syntax Description

speed	Integer that specifies the ring speed, either 4 for 4-Mbps or 16 for 16-Mbps operation.

Default

16-Mbps operation

Command Mode

Interface configuration

Usage Guidelines

Caution Configuring a ring speed that is wrong or incompatible with the connected Token Ring will cause the ring to beacon, which effectively takes the ring down and makes it nonoperational.

Example

The following example sets a Token Ring interface ring speed to 4 Mbps:

```
interface tokenring 0
ring-speed 4
```

scheduler-interval

To control the maximum amount of time that can elapse without running the lowest priority system processes, use the **scheduler-interval** global configuration command. Use the **no** form of this command to restore the default.

scheduler-interval *milliseconds*
no scheduler-interval

Syntax Description

milliseconds	An integer that specifies the interval in milliseconds. The minimum interval that you can specify is 500 milliseconds; there is no maximum value.

Default

The default is to allow high-priority operations to use as much of the central processor as needed.

Command Mode

Global configuration

Usage Guidelines

The normal operation of the network server allows the switching operations to use as much of the central processor as is required. If the network is running unusually heavy loads that do not allow the processor the time to handle the routing protocols, give priority to the system process scheduler.

Example

The following example changes the low-priority process schedule to an interval of 750 milliseconds:

```
scheduler-interval 750
```

show async-bootp

Use the **show async-bootp** EXEC command to display the parameters that have been configured for SLIP extended BootP requests.

show async-bootp

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Sample Display

The following is a sample output of the **show async-bootp** command:

```
cs# show async-bootp

The following extended data will be sent in BOOTP responses:

bootfile (for address 128.128.1.1) "pcboot"
bootfile (for address 131.108.1.111) "dirtboot"
subnet-mask 255.255.0.0
time-offset -3600
time-server 128.128.1.1
```

Table 1-6 describes significant fields shown in the display.

Table 1-6 Show Async-BootP Field Descriptions

Field	Description
bootfile... “pcboot”	Boot file for address 128.128.1.1 is named pcboot.
subnet-mask 255.255.0.0	Subnet mask.
time-offset -3600	Local time is one hour (3600 seconds) earlier than UTC time.
time-server 128.128.1.1	Address of the time server for the network.

Related Command

The dagger (†) indicates that the command is documented in another chapter of this manual.

async-bootp†

show async status

To list the status of asynchronous interfaces, use the **show async status** EXEC command:

show async status

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

Shows all asynchronous sessions, whether they are using SLIP or PPP encapsulation.

Sample Display

The following is sample output from the **show async status** command:

```
communication server> show async status
Async protocol statistics:
  Rcvd: 5448 packets, 7682760 bytes
        1 format errors, 0 checksum errors, 0 overrun, 0 no buffer
  Sent: 5455 packets, 7682676 bytes, 0 dropped


 Int           Local          Remote Qd InPack OutPac Inerr  Drops  MTU Qsz
   1     192.31.7.84         Dynamic  0      0      0     0      0 1500  10
```

Table 1-7 describes significant fields shown in the display.

Table 1-7 Asynchronous Statistics Display Field Descriptions

Field	Description
Rcvd:	Statistics on packets received.
5548 packets	Packets received.
7682760 bytes	Total number of bytes.
1 format errors	Packets with a bad IP header, even before the checksum is calculated.
0 checksum errors	Count of checksum errors.
0 overrun	Number of giants received.
0 no buffer	Number of packets received when no buffer was available.
Sent	Statistics on packets sent.
5455 packets	Packets sent.
7682676 bytes	Total number of bytes.
0 dropped	Number of packets dropped.
Int	Interface number.
*	Line currently in use.
Local	Local IP address on the link.

Field	Description
Remote	Remote IP address on the link; “Dynamic” indicates that a remote address is allowed but has not been specified; “None” indicates that no remote address is assigned or being used.
Qd	Number of packets on hold queue (Qsz is max).
InPack	Number of packets received.
OutPac	Number of packets sent.
Inerr	Number of total input errors; sum of format errors, checksum errors, overruns and no buffers.
Drops	Number of packets received that would not fit on the hold queue.
MTU	Current maximum transmission unit size.
Qsz	Current output hold queue size.

Related Commands

interface async

show controllers mci

Use the **show controllers mci** EXEC command to display all information under the Multiport Communications Interface card. This command displays information the system uses for bridging and routing that is specific to the interface hardware. The information displayed is generally useful for diagnostic tasks performed by technical support personnel only.

show controllers mci

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Sample Display

The following is sample output from the **show controllers mci** command:

```
communication server# show controllers mci

MCI 0, controller type 1.1, microcode version 1.8
   128 Kbytes of main memory, 4 Kbytes cache memory
22 system TX buffers, largest buffer size 1520
   Restarts: 0 line down, 0 hung output, 0 controller error
Interface 0 is Ethernet0, station address 0000.0c00.d4a6
   15 total RX buffers, 11 buffer TX queue limit, buffer size 1520
   Transmitter delay is 0 microseconds
Interface 1 is Serial0, electrical interface is V.35 DTE
   15 total RX buffers, 11 buffer TX queue limit, buffer size 1520
   Transmitter delay is 0 microseconds
   High speed synchronous serial interface
Interface 2 is Ethernet1, station address aa00.0400.3be4
   15 total RX buffers, 11 buffer TX queue limit, buffer size 1520
   Transmitter delay is 0 microseconds
Interface 3 is Serial1, electrical interface is V.35 DCE
   15 total RX buffers, 11 buffer TX queue limit, buffer size 1520
   Transmitter delay is 0 microseconds
   High speed synchronous serial interface
```

Table 1-8 describes significant fields shown in the display.

Table 1-8 Show Controllers MCI Field Descriptions

Field	Description
MCI 0	Card type and unit number (varies depending on card).
controller type 1.1	Version number of the card.
microcode version 1.8	Version number of the card's internal software (in read-only memory).
128 Kbytes of main memory	Amount of main memory on the card.
4 Kbytes cache memory	Amount of cache memory on the card.
22 system TX buffers	Number of buffers that hold packets to be transmitted.

Field	Description
largest buffer size 1520	Largest size of these buffers (in bytes).
Restarts 0 line down 0 hung output 0 controller error	Count of restarts due to the following conditions: Communication line down Output unable to transmit Internal error
Interface 0 is Ethernet0	Names of interfaces, by number.
electrical interface is V.35 DTE	Line interface type for serial connections.
15 total RX buffers	Number of buffers for received packets.
11 buffer TX queue limit	Maximum number of buffers in transmit queue.
Transmitter delay is 0 microseconds	Delay between outgoing frames.
Station address 0000.0c00.d4a6	Hardware address of the interface.

Note The interface type is only queried at startup. If the hardware changes *subsequent* to initial startup, then the wrong type is reported. This has *no* adverse effect on the operation of the software. For instance, if a DCE cable is connected to a dual-mode V.35 applique after the unit has been booted, then the display presented for **show interfaces** incorrectly reports attachment to a DTE device although the software recognizes the DCE interface and behaves accordingly.

Related Command

tx-queue-limit

show controllers token

To display information about memory management, error counters, and the CSC-1R, CSC-2R, and or CSC-R16M Token Ring interface cards, use the **show controllers token** prvileged EXEC command:

show controllers token

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

Depending on the board being used, the output can vary. This command also displays information that is proprietary to Cisco Systems. Thus, the information that **show controllers token** displays is of primary use to Cisco technical personnel. Information that is useful to users can be obtained using the **show interfaces tokenring** command, described later in this chapter.

Sample Display

The following is sample output on the AGS+ from the **show controllers token** command:

```
cs# show controllers token
TR Unit 0 is board 0 - ring 0

 state 3, dev blk: 0x1D2EBC, mailbox: 0x2100010, sca: 0x2010000
   current address: 0000.3080.6f40, burned in address: 0000.3080.6f40
   current TX ptr: 0xBA8, current RX ptr: 0x800

   Last Ring Status: none

 Stats: soft:0/0, hard:0/0, sig loss:0/0
        tx beacon: 0/0, wire fault 0/0, recovery: 0/0
        only station: 0/0, remote removal: 0/0
   Bridge: local 3330, bnum 1, target 3583
     max_hops 7, target idb: 0x0, not local
   Interface failures: 0  -- Bkgnd Ints: 0
   TX shorts 0, TX giants 0

   Monitor state: (active)
     flags 0xC0, state 0x0, test 0x0, code 0x0, reason 0x0
 f/w ver: 1.0, chip f/w: '000000.ME31100', [bridge capable]
     SMT versions: 1.01 kernel, 4.02 fastmac
     ring mode: F00, internal enables:  SRB REM RPS CRS/NetMgr
     internal functional: 0000011A (0000011A), group: 00000000 (00000000)
     if_state: 1, ints: 0/0, ghosts: 0/0, bad_states: 0/0
     t2m fifo purges: 0/0
     t2m fifo current: 0, t2m fifo max: 0/0, proto_errs: 0/0
     ring: 3330, bridge num: 1, target: 3583, max hops: 7
Packet counts:
       receive total:  298/6197, small: 298/6197, large 0/0
               runts: 0/0, giants: 0/0
               local: 298/6197, bridged: 0/0, promis: 0/0
             bad rif: 0/0, multiframe: 0/0
       ring num mismatch 0/0, spanning violations 0
```

```
       transmit total: 1/25, small: 1/25, large 0/0
                runts: 0/0, giants: 0/0, errors 0/0
bad fs: 0/0, bad ac: 0
congested: 0/0, not present: 0/0
     Unexpected interrupts: 0/0,  last unexp. int: 0

     Internal controller counts:
   line errors: 0/0,  internal errors: 0/0
   burst errors: 0/0,  ari/fci errors: 0/0
   abort errors: 0/0, lost frame: 0/0
   copy errors: 0/0, rcvr congestion: 0/0
   token errors: 0/0, frequency errors: 0/0
   dma bus errors: -/-, dma parity errors: -/-
     Internal controller smt state:
   Adapter MAC:     0000.3080.6f40, Physical drop:     00000000
   NAUN Address:    0000.a6e0.11a6, NAUN drop:         00000000
   Last source:     0000.a6e0.11a6, Last poll:         0000.3080.6f40
   Last MVID:       0006,           Last attn code:    0006
   Txmit priority:  0006,           Auth Class:        7FFF
   Monitor Error:   0000,           Interface Errors:  FFFF
   Correlator:      0000,           Soft Error Timer:  00C8
   Local Ring:      0000,           Ring Status:       0000
   Beacon rcv type: 0000,           Beacon txmit type: 0000
   Beacon type:     0000,           Beacon NAUN:       0000.a6e0.11a6
```

Table 1-9 describes the fields shown in the following line of sample output.

```
TR Unit 0 is board 0 - ring 0
```

Table 1-9 Show Controllers Token Field Descriptions—Part 1

Field	Description
TR Unit 0	Unit number assigned to the Token Ring interface associated with this output.
is board 0	Board number assigned to the Token Ring controller board associated with this interface.
ring 0	Number of the Token Ring associated with this board.

In the following output line, state 3 indicates the state of the board. The rest of this output line displays memory mapping that is of primary use to Cisco engineers.

```
state 3, dev blk: 0x1D2EBC, mailbox: 0x2100010, sca: 0x2010000
```

The following line also appears in **show interface token** output as the address and burned in address, respectively:

```
current address: 0000.3080.6f40, burned in address: 0000.3080.6f40
```

The following line of output displays buffer management pointers that change by board:

```
current TX ptr: 0xBA8, current RX ptr: 0x800
```

The following line of output indicates the ring status from the controller chip set. This information is used by LAN Network Manager:

```
Last Ring Status: none
```

The following lines of output show Token Ring statistics. See the Token Ring specification for more information.

```
Stats: soft:0/0, hard:0/0, sig loss:0/0
       tx beacon: 0/0, wire fault 0/0, recovery: 0/0
       only station: 0/0, remote removal: 0/0
```

The following line of output indicates that Token Ring communication has been enabled on the interface. If this line of output appears, the message "Source Route Bridge capable" should appear in the **show interfaces tokenring** display.

```
Bridge: local 3330, bnum 1, target 3583
```

Table 1-10 describes the fields shown in this line of sample output.

```
max_hops 7, target idb: 0x0, not local
```

Table 1-10 Show Controllers Token Field Descriptions—Part 2

Field	Description
max_hops 7	Maximum number of bridges.
target idb: 0x0	Destination interface definition.
not local	Indicates whether the interface has been defined as a local or remote bridge.

The following line of output is specific to the hardware:

```
Interface failures: 0  -- Bkgnd Ints: 0
```

In the following line of output, TX shorts are the number of packets the interface transmits that are discarded because they are smaller than the medium's minimum packet size. TX giants are the number of packets the interface transmits that are discarded because they exceed the medium's maximum packet size.

```
TX shorts 0, TX giants 0
```

The following line of output indicates the state of the controller. Possible values include active, failure, inactive, and reset:

```
Monitor state: (active)
```

The following line of output displays detailed information relating to the monitor state shown in the previous line of output. This information relates to the firmware on the controller. This information is relevant to Cisco engineers only if the monitor state is something other than active.

```
flags 0xC0, state 0x0, test 0x0, code 0x0, reason 0x0
```

Table 1-11 describes the fields in the following line of output:

```
f/w ver: 1.0 expr 0, chip f/w: '000000.ME31100', [bridge capable]
```

Table 1-11 Show Controllers Token Field Descriptions—Part 3

Field	Description
f/w ver: 1.0	Version of the Cisco firmware on the board.
chip f/w: '000000.ME31100'	Firmware on the chip set.
[bridge capable]	Interface has not been configured for bridging, but that it has that capability.

The following line of output displays the version numbers for the kernel and the accelerator microcode of the Madge firmware on the board; this firmware is the LLC interface to the chip set:

```
SMT versions: 1.01 kernel, 4.02 fastmac
```

The following line of output displays LAN Network Manager information that relates to ring status:

```
ring mode: F00, internal enables:  SRB REM RPS CRS/NetMgr
```

The following line of output corresponds to the functional address and the group address shown in **show interfaces tokenring** output:

```
internal functional: 0000011A (0000011A), group: 00000000 (00000000)
```

The following line of output displays interface board state information that is proprietary to Cisco Systems:

```
if_state: 1, ints: 0/0, ghosts: 0/0, bad_states: 0/0
```

The following output lines display information that is proprietary to Cisco Systems. Cisco engineers use this information for debugging purposes.

```
t2m fifo purges: 0/0
t2m fifo current: 0, t2m fifo max: 0/0, proto_errs: 0/0
```

Each of the fields in the following line of output maps to a field in the **show source bridge** display, as follows: ring maps to srn; bridge num maps to bn; target maps to trn; and max hops maps to max:

```
ring: 3330, bridge num: 1, target: 3583, max hops: 7
```

In the following lines of output, the number preceding the slash (/) indicates the count since the value was last displayed; the number following the slash (/) indicates count since the system was last booted:

```
Packet counts:
       receive total:  298/6197, small: 298/6197, large 0/0
```

In the following line of output, the number preceding the slash (/) indicates the count since the value was last displayed; the number following the slash (/) indicates count since the system was last booted. The runts and giants values that appear here correspond to the runts and giants values that appear in **show interfaces tokenring** output.

```
runts: 0/0, giants: 0/0
```

The following lines of output are receiver-specific information that Cisco engineers can use for debugging purposes:

```
local: 298/6197, bridged: 0/0, promis: 0/0
bad rif: 0/0, multiframe: 0/0
ring num mismatch 0/0, spanning violations 0
transmit total: 1/25, small: 1/25, large 0/0
runts: 0/0, giants: 0/0, errors 0/0
```

The following output lines include very specific statistics that are not relevant in most cases, but exist for historical purposes. In particular, the internal errors, burst errors, ari/fci, abort errors, copy errors, frequency errors, dma bus errors, and dma parity errors fields are not relevant.

```
     Internal controller counts:
line errors: 0/0,  internal errors: 0/0
burst errors: 0/0,  ari/fci errors: 0/0
abort errors: 0/0, lost frame: 0/0
copy errors: 0/0, rcvr congestion: 0/0
token errors: 0/0, frequency errors: 0/0
dma bus errors: -/-, dma parity errors: -/-
```

The following lines of output are low-level Token Ring interface statistics relating to the state and status of the Token Ring with respect to all other Token Rings on the line:

```
  Internal controller smt state:
Adapter MAC:     0000.3080.6f40, Physical drop:     00000000
NAUN Address:    0000.a6e0.11a6, NAUN drop:         00000000
Last source:     0000.a6e0.11a6, Last poll:         0000.3080.6f40
Last MVID:       0006,           Last attn code:    0006
Txmit priority:  0006,           Auth Class:        7FFF
Monitor Error:   0000,           Interface Errors:  FFFF
Correlator:      0000,           Soft Error Timer:  00C8
Local Ring:      0000,           Ring Status:       0000
Beacon rcv type: 0000,           Beacon txmit type: 0000
```

show interfaces

Use the **show interfaces** EXEC command to display statistics for all interfaces configured on the communication server. The resulting output varies, depending on the network for which an interface has been configured.

show interfaces [*interface-type unit*] [**accounting**]

Syntax Description

type unit	(Optional.) Specify that information for a particular interface controller be displayed. Allowed values for type include **async**, **ethernet**, **loopback**, **null**, **serial**, **tokenring**, and **tunnel**.
accounting	(Optional.) Displays the number of packets of each protocol type that has been sent through the interface. You can show these numbers for all interfaces, or you can specify a specific *type* and *unit*.

Command Mode

EXEC

Usage Guidelines

The **show interfaces** command displays statistics for the network interfaces. If you enter a **show interfaces** command for an interface type that has been removed from the communication server, interface statistics will be displayed accompanied by the following text: “Hardware has been removed.”

You will use the **show interfaces** command frequently while configuring and monitoring communication servers. The various forms of the **show interfaces** commands are described in detail in the sections immediately following this command.

To display the number of packets of each protocol type that have been sent through all configured interfaces, use the **show interfaces accounting** EXEC command. When you use the **accounting** option, only the accounting statistics are displayed.

Table 1-12 lists the protocols for which per-packet accounting information is kept.

Table 1-12 Per-Packet Counted Protocols

Protocol	Notes
ARP	For IP, Apollo, Frame Relay, SMDS.
DEC MOP	The communication servers use MOP packets to advertise their existence to DEC machines that use the MOP protocol. A communication server periodically broadcasts MOP packets to identify itself as a MOP host. This results in MOP packets being counted.
HP Probe	no note
IP	no note
Lan Manager	LAN Network Manager and IBM Network Manager.
Novell	no note

Sample Display

The following is sample output from the **show interfaces** command. Because your display will depend on the type and number of interface cards in your communication server, only a portion of the display is shown.

```
cs# show interfaces
Ethernet 0 is up, line protocol is up
  Hardware is MCI Ethernet, address is 0000.0c00.750c (bia 0000.0c00.750c)
  Internet address is 131.108.28.8, subnet mask is 255.255.255.0
  MTU 1500 bytes, BW 10000 Kbit, DLY 100000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 4:00:00
  Last input 0:00:00, output 0:00:00, output hang never
  Last clearing of "show interface" counters 0:00:00
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 2000 bits/sec, 4 packets/sec
     1127576 packets input, 447251251 bytes, 0 no buffer
     Received 354125 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     5332142 packets output, 496316039 bytes, 0 underruns
     0 output errors, 432 collisions, 0 interface resets, 0 restarts
---More---
```

Sample Display with Accounting Option

The following is sample output from the **show interfaces accounting** command:

```
cs# show interfaces accounting

Ethernet0
                Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                      IP     873171  735923409      34624    9644258
                  Novell     163849   12361626      57143    4272468
                 DEC MOP          0          0          1         77
                     ARP      69618    4177080       1529      91740
```

When the output indicates an interface is “disabled,” the communication server has received excessive errors (over 5000 in a keepalive period).

show interfaces async

Use the **show interfaces async** EXEC command to display information about the serial interface.

show interfaces async *unit* [**accounting**]

Syntax Description

unit	Must be 1.
accounting	(Optional.) Displays the number of packets of each protocol type that have been sent through the interface.

Command Mode

EXEC

Sample Display

The following is sample output from the **show interfaces async** command:

```
cs# show interfaces async 1

Async 1 is up, line protocol is up
   Hardware is Async Serial
Internet address is 1.0.0.1, subnet mask is 255.0.0.0
MTU 1500 bytes, BW 9 Kbit, DLY 100000 usec, rely 255/255, load 56/255
Encapsulation SLIP, keepalive set (0 sec)
Last input 0:00:03, output 0:00:03, output hang never
Last clearing of “show interface” counters never
Output queue 0/3, 2 drops; input queue 0/0, 0 drops
Five minute input rate 0 bits/sec, 1 packets/sec
Five minute output rate 2000 bits/sec, 1 packets/sec
273 packets input, 13925 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
221 packets output, 41376 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets, 0 restarts
0 carrier transitions
```

Note The communication server does not detect the start of a SLIP packet on an asynchronous interface. When the SLIP connection is established, the communication server only knows that it is working with the SLIP host that is generating traffic after it receives an END byte. Data that is transmitted before the END byte is counted as an error.

Table 1-13 describes the fields shown in the display.

Table 1-13 Show Interfaces Async Field Descriptions

Field	Description
Async... is {up | down} ...is administratively down	Indicates whether the interface hardware is currently active (whether carrier detect is present) and if it has been taken down by an administrator.
line protocol is {up | down | administratively down}	Indicates whether the software processes that handle the line protocol think the line is usable (that is, whether keepalives are successful).
Hardware is	Hardware type.
Internet address is	IP address and subnet mask, followed by packet size.
MTU	Maximum Transmission Unit of the interface.
BW	Bandwidth of the interface in kilobits per second.
DLY	Delay of the interface in microseconds.
rely	Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over five minutes.
load	Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over five minutes. The calculation uses the value from the **bandwidth** interface configuration command.
Encapsulation	Encapsulation method assigned to interface.
keepalive	Indicates whether keepalives are set or not.
Last input	Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed.
Last output	The number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface.
output hang	Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the "last" fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed.
Last clearing	The time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared. *** indicates the elapsed time is too large to be displayed. 0:00:00 indicates the counters were cleared more than 2^{31}ms (and less than 2^{32}ms) ago.
Output queue, drops input queue, drops	Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue.
Five minute input rate, Five minute output rate	Average number of bits and packets transmitted per second in the last five minutes.
packets input	Total number of error-free packets received by the system.
bytes input	Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system.

Field	Description
no buffers	Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events.
broadcasts	Total number of broadcast or multicast packets received by the interface.
runts	Number of packets that are discarded because they are smaller than the medium's minimum packet size.
giants	Number of packets that are discarded because they exceed the medium's maximum packet size.
input errors	Total number of no buffer, runts, giants, CRCs, frame, overrun, ignored, and abort counts. Other input-related errors can also increment the count, so that this sum may not balance with the other counts.
CRC	The Cyclic Redundancy Checksum generated by the originating LAN station or far end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRC's is usually the result of collisions or a station transmitting bad data. On a serial link, CRC's usually indicate noise, gain hits or other transmission problems on the data link.
frame	Number of packets received incorrectly having a CRC error and a noninteger number of octets. On a serial line, this is usually the result of noise or other transmission problems.
overrun	Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data.
ignored	Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be incremented.
abort	Illegal sequence of one bits on a serial interface. This usually indicates a clocking problem between the serial interface and the data link equipment.
packets output	Total number of messages transmitted by the system.
bytes	Total number of bytes, including data and MAC encapsulation, transmitted by the system.
output errors	Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the enumerated output errors, as some datagrams might have more than one error, and others might have errors that do not fall into any of the specifically tabulated categories.

Field	Description
interface resets	Number of times an interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds. On a serial line, this can be caused by a malfunctioning modem that is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a serial interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when an interface is looped back or shut down.
restarts	Number of times the controller was restarted because of errors.
carrier transitions	Number of times the carrier detect signal of a serial interface has changed state. Indicates modem or line problems if the carrier detect line is changing state often.
Protocol	Protocol that is operating on the interface.
Pkts In	Number of packets received for that protocol.
Chars In	Number of characters received for that protocol.
Pkts Out	Number of packets transmitted for that protocol.
Chars Out	Number of characters transmitted for that protocol.

Sample Display with Accounting Option

The following is a sample display from the **show interfaces async accounting** command:

```
cs# show interfaces async 0 accounting

Async 0
   Protocol  Pkts In   Chars In  Pkts Out   Chars Out
   IP        7344      4787842   1803       1535774
   DEC MOP   0         0         127        9779
   ARP       7         420       39         2340
```

The **show line** and **show slip** commands can also be useful in monitoring asynchronous interfaces.

show interfaces brief

To list a brief summary of an interface IP information and status, use the **show interfaces brief** EXEC command.

show interfaces brief

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Sample Display

The following provides sample output from the **show interfaces brief** command:

```
cs# show interfaces brief
Any interface listed with OK? value "NO" does not have a valid configuration
Interface    IP-Address      OK?  Method     Status                Protocol
Ethernet0    131.108.160.22  YES  NVRAM      up                    up
```

Related Command

show interfaces

show interfaces dialer

Use the **show interfaces dialer** EXEC command to display information about the dialer interface.

show interfaces dialer *unit* [**accounting**]

Syntax Description

unit	Must match a port number on the selected interface.
accounting	(Optional.) Displays the number of packets of each protocol type that have been sent through the interface.

Command Mode

EXEC

show interfaces ethernet

Use the **show interfaces ethernet** EXEC command to display information about an Ethernet interface on the communication server.

show interfaces ethernet *unit* [**accounting**]

Syntax Description

unit	Must match a port number on the selected interface.
accounting	(Optional.) Displays the number of packets of each protocol type that have been sent through the interface.

Command Mode

EXEC

Usage Guidelines

If you do not provide values for the parameter *unit*, or *type*, *slot*, and *port*, the command will display statistics for all network interfaces. The optional keyword **accounting** displays the number of packets of each protocol type that have been sent through the interface.

Sample Display

The following is sample output from the **show interfaces** command for the Ethernet 0 interface:

```
cs# show interfaces ethernet 0

Ethernet 0 is up, line protocol is up
   Hardware is MCI Ethernet, address is aa00.0400.0134 (bia 0000.0c00.4369)
   Internet address is 131.108.1.1, subnet mask is 255.255.255.0
   MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
   Encapsulation ARPA, loopback not set, keepalive set (10 sec)
   ARP type: ARPA, PROBE, ARP Timeout 4:00:00
   Last input 0:00:00, output 0:00:00, output hang never
   Output queue 0/40, 0 drops; input queue 0/75, 2 drops
   Five minute input rate 61000 bits/sec, 4 packets/sec
   Five minute output rate 1000 bits/sec, 2 packets/sec
       2295197 packets input, 305539992 bytes, 0 no buffer
       Received 1925500 broadcasts, 0 runts, 0 giants
       3 input errors, 3 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
       3594664 packets output, 436549843 bytes, 0 underruns
       8 output errors, 1790 collisions, 10 interface resets, 0 restarts
```

Table 1-14 describes significant fields shown in the display.

Table 1-14 Show Interfaces Ethernet Field Descriptions

Field	Description
Ethernet ... is up ...is administratively down	Indicates whether the interface hardware is currently active and if it has been taken down by an administrator. “Disabled” indicates the communication server has received over 5000 errors in a keepalive interval, which is 10 seconds by default.
line protocol is {up | down | administratively down}	Indicates whether the software processes that handle the line protocol believe the interface is usable (that is, whether keepalives are successful) or if it has been taken down by an administrator.
Hardware	Hardware type (for example, MCI Ethernet, SCI, Ethernet) and address.
Internet address	IP address followed by subnet mask.
MTU	Maximum Transmission Unit of the interface.
BW	Bandwidth of the interface in kilobits per second.
DLY	Delay of the interface in microseconds.
rely	Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over five minutes.
load	Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over five minutes.
Encapsulation	Encapsulation method assigned to interface.
ARP type:	Type of Address Resolution Protocol assigned.
loopback	Indicates whether loopback is set or not.
keepalive	Indicates whether keepalives are set or not.
Last input	Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed.
Last output	Number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface.
output	Number of hours, minutes, and seconds since the last packet was successfully transmitted by the interface. Useful for knowing when a dead interface failed.
output hang	Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the “last” fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed.
Last clearing	Time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared. *** indicates the elapsed time is too large to be displayed. 0:00:00 indicates the counters were cleared more than 2^{31}ms (and less than 2^{32}ms) ago.
Output queue, input queue, drops	Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue.

Field	Description
Five minute input rate, Five minute output rate	Average number of bits and packets transmitted per second in the last five minutes. If the interface is not in promiscuous mode, it senses network traffic it sends and receives (rather than all network traffic). The five-minute input and output rates should be used only as an approximation of traffic per second during a given five-minute period. These rates are exponentially weighted averages with a time constant of five minutes. A period of four time constants must pass before the average will be within two percent of the instantaneous rate of a uniform stream of traffic over that period.
packets input	Total number of error-free packets received by the system.
bytes input	Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system.
no buffers	Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events.
Received ... broadcasts	Total number of broadcast or multicast packets received by the interface.
runts	Number of packets that are discarded because they are smaller than the medium's minimum packet size. For instance, any Ethernet packet that is less than 64 bytes is considered a runt.
giants	Number of packets that are discarded because they exceed the medium's maximum packet size. For example, any Ethernet packet that is greater than 1,518 bytes is considered a giant.
input error	Includes runts, giants, no buffer, CRC, frame, overrun, and ignored counts. Other input-related errors can also cause the input errors count to be increased, and some datagrams may have more than one error; therefore, this sum may not balance with the sum of enumerated input error counts.
CRC	Cyclic Redundancy Checksum generated by the originating LAN station or far-end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRCs is usually the result of collisions or a station transmitting bad data.
frame	Number of packets received incorrectly having a CRC error and a noninteger number of octets. On a LAN, this is usually the result of collisions or a malfunctioning Ethernet device.
overrun	Number of times the receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data.
ignored	Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be increased.
packets output	Total number of messages transmitted by the system.
bytes	Total number of bytes, including data and MAC encapsulation, transmitted by the system.

Field	Description
underruns	Number of times that the transmitter has been running faster than the communication server can handle. This might never be reported on some interfaces.
output errors	Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this might not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories.
collisions	Number of messages retransmitted due to an Ethernet collision. This is usually the result of an overextended LAN (Ethernet or transceiver cable too long, more than two repeaters between stations, or too many cascaded multiport transceivers). A packet that collides is counted only once in output packets.
interface resets	Number of times an interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds' time. On a serial line, this can be caused by a malfunctioning modem that is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a serial interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when an interface is looped back or shut down.
restarts	Number of times a Type 2 Ethernet controller was restarted because of errors.

show interfaces loopback

Use the **show interfaces loopback** EXEC command to display information about the dialer interface.

show interfaces loopback *unit* [**accounting**]

Syntax Description

unit	Must match a port number on the selected interface.
accounting	(Optional.) Displays the number of packets of each protocol type that have been sent through the interface.

Command Mode

EXEC

Sample Displays

The following is sample output from the **show interfaces loopback** command:

```
cs# show int loop 0
Loopback0 is up, line protocol is up
  Hardware is Loopback
  MTU 1500 bytes, BW 1 Kbit, DLY 50 usec, rely 255/255, load 1/255
  Encapsulation UNKNOWN, loopback not set, keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Output queue 0/0, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets, 0 restarts
```

The following is sample when the **accounting** keyword is included:

```
cs# show int loop 0 acc
Loopback0
               Protocol    Pkts In   Chars In   Pkts Out  Chars Out
No traffic sent or received on this interface.
communication server#
```

Table 1-15 describes significant fields shown in the displays.

Table 1-15 Show Interfaces Loopback Descriptions

Field	Description
Loopback is {up | down} ...is administratively down	Indicates whether the interface hardware is currently active (whether carrier detect is present) and if it has been taken down by an administrator. "Disabled" indicates the communication server has received over 5000 errors in a keepalive interval, which is 10 seconds by default.
line protocol is {up | down | administratively down}	Indicates whether the software processes that handle the line protocol considers the line usable (that is, whether keepalives are successful).
Hardware	Hardware is Loopback.
MTU	Maximum Transmission Unit of the interface.
BW	Bandwidth of the interface in kilobits per second.
DLY	Delay of the interface in microseconds.
rely	Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over five minutes.
load	Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over five minutes.
Encapsulation	Encapsulation method assigned to interface.
loopback	Indicates whether loopback is set and type of loopback test.
keepalive	Indicates whether keepalives are set or not.
Last input	Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed.
Last output	Number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface.
output hang	Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the "last" fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed.
Last clearing	Time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared. *** indicates the elapsed time is too large to be displayed. 0:00:00 indicates the counters were cleared more than 2^{31}ms (and less than 2^{32}ms) ago.
Output queue, drops Input queue, drops	Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue.
Five minute input rate, Five minute output rate	Average number of bits and packets transmitted per second in the last five minutes.
packets input	Total number of error-free packets received by the system.

Field	Description
bytes input	Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system.
no buffer	Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events.
broadcasts	Total number of broadcast or multicast packets received by the interface.
runts	Number of packets that are discarded because they are smaller than the medium's minimum packet size.
giants	Number of packets that are discarded because they exceed the medium's maximum packet size.
input errors	Sum of all errors that prevented the receipt of datagrams on the interface being examined. This may not balance with the sum of the enumerated output errors, because some datagrams may have more than one error and others may have errors that do not fall into any of the specifically tabulated categories.
CRC	Cyclic Redundancy Checksum generated by the originating LAN station or far-end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRCs is usually the result of collisions or a station transmitting bad data. On a serial link, CRCs usually indicate noise, gain hits, or other transmission problems on the data link. CRC errors are also reported when a far-end abort occurs, and when the idle flag pattern is corrupted. This makes it possible to get CRC errors even when there is no data traffic.
frame	Number of packets received incorrectly having a CRC error and a noninteger number of octets. On a serial line, this is usually the result of noise or other transmission problems.
overrun	Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data.
ignored	Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be increased.
abort	Number of packets whose receipt was aborted.
input packets with dribble condition detected	
packets output	Total number of messages transmitted by the system.
bytes output	Total number of bytes, including data and MAC encapsulation, transmitted by the system.

Field	Description
underruns	Number of times that the far-end transmitter has been running faster than the near-end communication server's receiver can handle. This might never happen (be reported) on some interfaces.
output errors	Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this might not balance with the sum of the enumerated output errors, as some datagrams might have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories.
collisions	
interface resets	Number of times an interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds time. On a serial line, this can be caused by a malfunctioning modem that is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a serial interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when an interface is looped back or shut down.
restarts	Number of times the controller was restarted because of errors.
Protocol	Protocol that is operating on the interface.
Pkts In	Number of packets received for that protocol.
Chars In	Number of characters received for that protocol.
Pkts Out	Number of packets transmitted for that protocol.
Chars Out	Number of characters transmitted for that protocol.

show interfaces serial

Use the **show interfaces serial** command to display information about a serial interface.

show interfaces serial *unit* [**accounting**]

Syntax Description

unit	Must match the interface port number.
accounting	(Optional.) Displays the number of packets of each protocol type that have been sent through the interface.

Command Mode

EXEC

Sample Display

The following is sample output from the **show interfaces** command for a synchronous serial interface:

```
cs# show interfaces serial
Serial 0 is up, line protocol is up
   Hardware is MCI Serial
   Internet address is 150.136.190.203, subnet mask is 255.255.255.0
   MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
   Last input 0:00:07, output 0:00:00, output hang never
   Output queue 0/40, 0 drops; input queue 0/75, 0 drops
   Five minute input rate 0 bits/sec, 0 packets/sec
   Five minute output rate 0 bits/sec, 0 packets/sec
       16263 packets input, 1347238 bytes, 0 no buffer
       Received 13983 broadcasts, 0 runts, 0 giants
       2 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 2 abort
1 carrier transitions

     22146 packets output, 2383680 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets, 0 restarts
```

Table 1-16 describes significant fields shown in the display.

Table 1-16 Show Interfaces Serial Field Descriptions

Field	Description
Serial ... is {up | down} ...is administratively down	Indicates whether the interface hardware is currently active (whether carrier detect is present) and if it has been taken down by an administrator. “Disabled” indicates the communication server has received over 5000 errors in a keepalive interval, which is 10 seconds by default.
line protocol is {up | down}	Indicates whether the software processes that handle the line protocol consider the line usable (that is, whether keepalives are successful) or if it has been taken down by an administrator.
Hardware is	Specifies the hardware type.
Internet address is	Specifies the Internet address and subnet mask.
MTU	Maximum Transmission Unit of the interface.

Field	Description
BW 1544 Kbit	Indicates the value of the bandwidth parameter that has been configured for the interface (in kilobits per second). The bandwidth parameter is used to compute IGRP metrics only. If the interface is attached to a serial line with a line speed that does not match the default (1536 or 1544 for T1 and 56 for a standard synchronous serial line), use the **bandwidth** command to specify the correct line speed for this serial line.
DLY	Delay of the interface in microseconds.
rely	Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over five minutes.
load	Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over five minutes.
Encapsulation	Encapsulation method assigned to interface.
loopback	Indicates whether loopback is set or not.
keepalive	Indicates whether keepalives are set or not.
Last input	Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed.
Last output	Number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface.
output hang	Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the "last" fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed.
Output queue, drops input queue, drops	Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue.
Five minute input rate Five minute output rate	Average number of bits and packets transmitted per second in the last five minutes. The five-minute input and output rates should be used only as an approximation of traffic per second during a given five-minute period. These rates are exponentially weighted averages with a time constant of five minutes. A period of four time constants must pass before the average will be within two percent of the instantaneous rate of a uniform stream of traffic over that period.
packets input	Total number of error-free packets received by the system.
bytes input	Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system.
no buffers	Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events.
Received ... broadcasts	Total number of broadcast or multicast packets received by the interface.
runts	Number of packets that are discarded because they are smaller than the medium's minimum packet size.

Field	Description
giants	Number of packets that are discarded because they exceed the medium's maximum packet size.
input error	Total number of no buffer, runts, giants, CRCs, frame, overrun, ignored, and abort counts. Other input-related errors can also increment the count, so that this sum may not balance with the other counts.
CRC	Cyclic Redundancy Checksum generated by the originating station or far-end device does not match the checksum calculated from the data received. On a serial link, CRCs usually indicate noise, gain hits, or other transmission problems on the data link.
frame	Number of packets received incorrectly having a CRC error and a noninteger number of octets. On a serial line, this is usually the result of noise or other transmission problems.
overrun	Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data.
ignored	Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. Broadcast storms and bursts of noise can cause the ignored count to be increased.
abort	Illegal sequence of one bits on a serial interface. This usually indicates a clocking problem between the serial interface and the data link equipment.
packets output	Total number of messages transmitted by the system.
bytes output	Total number of bytes, including data and MAC encapsulation, transmitted by the system.
underruns	Number of times that the transmitter has been running faster than the communication server can handle. This may never be reported on some interfaces.
output errors	Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this might not balance with the sum of the enumerated output errors, as some datagrams may have more than one error, and others might have errors that do not fall into any of the specifically tabulated categories.
collisions	Number of messages retransmitted due to an Ethernet collision. This usually is the result of an overextended LAN (Ethernet or transceiver cable too long, more than two repeaters between stations, or too many cascaded multiport transceivers). Some collisions are normal. However, if your collision rate climbs to around 4-5%, you should consider verifying that there is no faulty equipment on the segment and/or moving some existing stations to a new segment. A packet that collides is counted only once in output packets.

Field	Description
interface resets	Number of times an interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds' time. On a serial line, this can be caused by a malfunctioning modem that is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a serial interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when an interface is looped back or shut down.
restarts	Number of times the controller was restarted because of errors.
carrier transitions	Number of times the carrier detect signal of a serial interface has changed state. Indicates modem or line problems if the carrier detect line is changing state often.

Sample Display with Frame Relay Encapsulation

When using the frame relay encapsulation, use the **show interfaces** command to display information on the multicast DLCI, the DLCI of the interface, and the LMI DLCI used for the local management interface.

The multicast DLCI and the local DLCI can be set using the **frame-relay multicast-dlci** and the **frame-relay local-dlci** configuration commands, or provided through the local management interface. The status information is taken from the LMI, when active.

The following is a sample display from the **show interfaces serial** output when using frame relay encapsulation.

```
cs# show interfaces serial

Serial 2 is up, line protocol is up
   Hardware type is MCI Serial
   Internet address is 131.108.122.1, subnet mask is 255.255.255.0
   MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
   Encapsulation FRAME-RELAY, loopback not set, keepalive set (10 sec)
   multicast DLCI 1022,  status defined, active
   source DLCI    20, status defined, active
   LMI DLCI 1023, LMI sent 10, LMI stat recvd 10, LMI upd recvd 2
   Last input 7:21:29, output 0:00:37, output hang never
   Output queue 0/100, 0 drops; input queue 0/75, 0 drops
   Five minute input rate 0 bits/sec, 0 packets/sec
   Five minute output rate 0 bits/sec, 0 packets/sec
       47 packets input, 2656 bytes, 0 no buffer
       Received 5 broadcasts, 0 runts, 0 giants
       5 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 57 abort
       518 packets output, 391205 bytes
       0 output errors, 0 collisions, 0 interface resets, 0 restarts
       1 carrier transitions
```

In this display, the multicast DLCI has been changed to 1022 with the **frame-relay multicast-dlci** interface configuration command.

The display shows, the statistics for the LMI are the number of status inquiry messages sent (LMI sent), the number of status messages received (LMI recvd), and the number of status updates received (upd recvd). See the *Frame Relay Interface* specification for additional explanations of this output.

Sample Display with ANSI LMI

For a serial interface with the ANSI LMI enabled, use the **show interfaces** command to determine the LMI type implemented.

The following is a sample display from the **show interfaces** output for a serial interface with the ANSI LMI enabled.

```
cs# show interfaces serial

Serial 1 is up, line protocol is up
   Hardware is MCI Serial
   Internet address is 131.108.121.1, subnet mask is 255.255.255.0
   MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
   Encapsulation FRAME-RELAY, loopback not set, keepalive set
   LMI DLCI    0, LMI sent 10, LMI stat recvd 10
   LMI type is ANSI Annex D
   Last input 0:00:00, output 0:00:00, output hang never
   Output queue 0/40, 0 drops; input queue 0/75, 0 drops
   Five minute input rate 0 bits/sec, 1 packets/sec
   Five minute output rate 1000 bits/sec, 1 packets/sec
       261 packets input, 13212 bytes, 0 no buffer
       Received 33 broadcasts, 0 runts, 0 giants
       0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
       238 packets output, 14751 bytes, 0 underruns
       0 output errors, 0 collisions, 0 interface resets, 0 restarts
```

Notice that the **show interfaces** output for a serial interface with ANSI LMI shown in this display is very similar to that for encapsulation set to Frame Relay, as shown in the previous display. Table 1-17 describes the few differences that exist.

Table 1-17 Show Interfaces Field Description with ANSI LMI

Field	Description
LMI DLCI 0	Identifies the DLCI used by the LMI for this interface. Default: 1023.
LMI sent 10	Number of LMI packets the communication server sent.
LMI type is ANSI Annex D	Indicates that the interface is configured for the ANS-adopted Frame Relay specification T1.617 Annex D.

Sample Display with LAPB Encapsulation

Use the **show interfaces** command to display operation statistics for an interface using LAPB encapsulation.

The following is sample output from the **show interfaces** command for a serial interface using LAPB encapsulation:

```
communication server# show interfaces

LAPB state is DISCONNECT, T1 3000, N1 12000, N2 20, K7, TH 3000
Window is closed
IFRAMEs 12/28 RNRs 0/1 REJs 13/1 SABMs 1/13 FRMRs 3/0 DISCs 0/11
```

Table 1-18 shows the fields relevant to all LAPB connections.

Table 1-18 Show Interfaces Serial Fields and Descriptions When LAPB is Enabled

Parameter	Description
LAPB state is DISCONNECT	State of the LAPB protocol.
T1 3000, N1 12000, ...	Current parameter settings.
Window is closed	Indicates that no more frames can be transmitted until some outstanding frames have been acknowledged.
IFRAMEs 12/28 RNRs 0/1 ...	Count of the different types of frames in the form of sent/ received.

Show Interfaces Serial with PPP

An interface configured for synchronous PPP encapsulation differs from the standard **show interface serial** output in the fourth and fifth lines displayed. An interface configured for PPP might include the following information.

```
Encapsulation PPP, loopback not set, keepalive set (10 sec)
PPP: No valid link quality reports received.
```

The output line that reads “PPP: No valid link quality reports received” indicates that no reports have been received. If link quality monitoring is not negotiated, then that line will indicate:

```
PPP: LQM not negotiated.
```

If link quality monitoring has been negotiated, and if link quality reports have been received, it will display:

```
PPP: LQR transmit interval 10 sec, receive interval 10 sec
  local tx/remote rx: packets 50/50  bytes 147/147  success 16/16
   remote tx/local rx:  packets 49/50  bytes 753/790  success 16/16
```

This display contrasts the number of packets and bytes transmitted with the number received by the remote end, and the number of successful link quality reports received.

show interfaces tokenring

Use the **show interfaces tokenring** EXEC command to display information about the Token Ring interface and the state of source route bridging.

show interfaces tokenring *unit* [**accounting**]

Syntax Description

unit	Must match the interface port line number.
accounting	(Optional.) Displays the number of packets of each protocol type that have been sent through the interface.

Command Mode

EXEC

Usage Guidelines

If you do not provide values for the parameters *slot*, and *port*, the command will display statistics for all the network interfaces. The optional keyword **accounting** displays the number of packets of each protocol type that have been sent through the interface.

Sample Display

The following is sample output from the **show interfaces tokenring** command.

```
cs# show interfaces tokenring

TokenRing 0 is up, line protocol is up
Hardware is 16/4 Token Ring, address is 5500.2000.dc27 (bia 0000.3000.072b)
   Internet address is 150.136.230.203, subnet mask is 255.255.255.0
   MTU 8136 bytes, BW 16000 Kbit, DLY 630 usec, rely 255/255, load 1/255
   Encapsulation SNAP, loopback not set, keepalive set (10 sec)
   ARP type: SNAP, ARP Timeout 4:00:00
   Ring speed: 16 Mbps
   Single ring node, Source Route Bridge capable
   Group Address: 0x00000000, Functional Address: 0x60840000
   Last input 0:00:01, output 0:00:01, output hang never
   Output queue 0/40, 0 drops; input queue 0/75, 0 drops
   Five minute input rate 0 bits/sec, 0 packets/sec
   Five minute output rate 0 bits/sec, 0 packets/sec
   16339 packets input, 1496515 bytes, 0 no buffer
        Received 9895 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     32648 packets output, 9738303 bytes, 0 underruns
0 output errors, 0 collisions, 2 interface resets, 0 restarts
     5 transitions
```

Table 1-19 describes significant fields shown in the display.

Table 1-19 Show Interfaces Tokenring Field Descriptions

Field	Description
Token Ring is up | down	Interface is either currently active and inserted into ring (up) or inactive and not inserted (down). "Disabled" indicates the communication server has received over 5000 errors in a keepalive interval, which is 10 seconds by default.
Token Ring is Reset	Hardware error has occurred.
Token Ring is Initializing	Hardware is up, in the process of inserting the ring.
Token Ring is Administratively Down	Hardware has been taken down by an administrator.
line protocol is {up | down | administratively down}	Indicates whether the software processes that handle the line protocol believe the interface is usable (that is, whether keepalives are successful).
Hardware	Hardware type. "Hardware is Token Ring" indicates that the board is a CSC-R board. "Hardware is 16/4 Token Ring" indicates that the board is a CSC-R16 board. Also shows the address of the interface.
Internet address	Lists the IP address followed by subnet mask.
MTU	Maximum Transmission Unit of the interface.
BW	Bandwidth of the interface in kilobits per second.
DLY	Delay of the interface in microseconds.
rely	Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over five minutes.
load	Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over five minutes.
Encapsulation	Encapsulation method assigned to interface.
loopback	Indicates whether loopback is set or not.
keepalive	Indicates whether keepalives are set or not.
ARP type:	Type of Address Resolution Protocol assigned.
Ring speed:	Speed of Token Ring—4 or 16 Mbps.
{Single ring/multiring node}	Indicates whether a node is enabled to collect and use source routing information (RIF) for routable Token Ring protocols.
Group Address:	Interface's group address, if any. The group address is a multicast address; any number of interfaces on the ring may share the same group address. Each interface may have at most one group address.
Last input	Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed.
Last output	Number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface.

Field	Description
output hang	Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the "last" fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed.
Last clearing	Time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared. *** indicates the elapsed time is too large to be displayed. 0:00:00 indicates the counters were cleared more than 2^{31}ms (and less than 2^{32}ms) ago.
Output queue, drops Input queue, drops	Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue.
Five minute input rate, Five minute output rate	Average number of bits and packets transmitted per second in the last five minutes. The five-minute input and output rates should be used only as an approximation of traffic per second during a given five-minute period. These rates are exponentially weighted averages with a time constant of five minutes. A period of four time constants must pass before the average will be within two percent of the instantaneous rate of a uniform stream of traffic over that period.
packets input	Total number of error-free packets received by the system.
bytes input	Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system.
no buffers	Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events.
broadcasts	Total number of broadcast or multicast packets received by the interface.
runts	Number of packets that are discarded because they are smaller than the medium's minimum packet size.
giants	Number of packets that are discarded because they exceed the medium's maximum packet size.
CRC	Cyclic Redundancy Checksum generated by the originating LAN station or far-end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRCs is usually the result of a station transmitting bad data.
frame	Number of packets received incorrectly having a CRC error and a noninteger number of octets.
overrun	Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data.

Field	Description
ignored	Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be increased.
packets output	Total number of messages transmitted by the system.
bytes output	Total number of bytes, including data and MAC encapsulation, transmitted by the system.
underruns	Number of times that the far-end transmitter has been running faster than the near-end communication server's receiver can handle. This may never be reported on some interfaces.
output errors	Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this might not balance with the sum of the enumerated output errors, as some datagrams might have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories.
collisions	Since a Token Ring cannot have collisions, this statistic is nonzero only if an unusual event occurred when frames were being queued or dequeued by the system software.
interface resets	Number of times an interface has been reset. The interface may be reset by the administrator or automatically when an internal error occurs.
Restarts	Should always be zero for Token Ring interfaces.
transitions	Number of times the ring made a transition from up to down, or vice versa. A large number of transitions indicates a problem with the ring or the interface.

show interfaces tunnel

To list tunnel interface information, use the **show interfaces tunnel** EXEC command.

show interfaces tunnel *unit* [**accounting**]

Syntax Description

unit	Must match the interface port line number.
accounting	(Optional.) Displays the number of packets of each protocol type that have been sent through the interface.

Command Mode

EXEC

Sample Display

The following example provides sample output from the **show interface tunnel** command:

```
cs# show interfaces tunnel 4
Tunnel4 is up, line protocol is down
  Hardware is Routing Tunnel
  MTU 1500 bytes, BW 9 Kbit, DLY 500000 usec, rely 255/255, load 1/255
  Encapsulation TUNNEL, loopback not set, keepalive set (10 sec)
  Tunnel source 0.0.0.0, destination 0.0.0.0
  Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Output queue 0/0, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets, 0 restarts
```

Table 1-20 describes significant fields shown in the display.

Table 1-20 Show Interfaces Tunnel Field Descriptions

Field	Description
Tunnel is up | down	Interface is currently active and inserted into ring (up) or inactive and not inserted (down).
line protocol is {up | down | administratively down}	Shows line protocol up if a valid route is available to the tunnel destination. Shows line protocol down if no route is available, or if the route would be recursive.
Hardware	Specifies the hardware type.
MTU	Maximum Transmission Unit of the interface.
BW	Bandwidth of the interface in kilobits per second.
DLY	Delay of the interface in microseconds.

Field	Description
rely	Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over five minutes.
load	Load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over five minutes.
Encapsulation	Encapsulation method is always TUNNEL for tunnels.
loopback	Indicates whether loopback is set or not.
keepalive	Indicates whether keepalives are set or not.
Tunnel source	IP address used as the source address for packets in the tunnel.
destination	IP address of the host destination.
Tunnel protocol	Tunnel transport protocol (the protocol the tunnel is using). This is based on the **tunnel mode** command, which defaults to GRE.
key	ID key for the tunnel interface, unless disabled.
sequencing	Indicates whether the tunnel interface drops datagrams that arrive out of order. Can be disabled.
Last input	Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Useful for knowing when a dead interface failed.
Last output	Number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface.
output hang	Number of hours, minutes, and seconds (or never) since the interface was last reset because of a transmission that took too long. When the number of hours in any of the "last" fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed.
Last clearing	Time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared. *** indicates the elapsed time is too large to be displayed. 0:00:00 indicates the counters were cleared more than 2^{31}ms (and less than 2^{32}ms) ago.
Output queue, drops Input queue, drops	Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue.

Field	Description
Five minute input rate, Five minute output rate	Average number of bits and packets transmitted per second in the last five minutes. The five-minute input and output rates should be used only as an approximation of traffic per second during a given five-minute period. These rates are exponentially weighted averages with a time constant of five minutes. A period of four time constants must pass before the average will be within two percent of the instantaneous rate of a uniform stream of traffic over that period.
packets input	Total number of error-free packets received by the system.
bytes input	Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system.
no buffers	Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events.
broadcasts	Total number of broadcast or multicast packets received by the interface.
runts	Number of packets that are discarded because they are smaller than the medium's minimum packet size.
giants	Number of packets that are discarded because they exceed the medium's maximum packet size.
CRC	Cyclic Redundancy Checksum generated by the originating LAN station or far-end device does not match the checksum calculated from the data received. On a LAN, this usually indicates noise or transmission problems on the LAN interface or the LAN bus itself. A high number of CRCs is usually the result of a station transmitting bad data.
frame	Number of packets received incorrectly having a CRC error and a noninteger number of octets.
overrun	Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data.
ignored	Number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be increased.
abort	Illegal sequence of one bits on a serial interface. This usually indicates a clocking problem between the serial interface and the data link equipment.
packets output	Total number of messages transmitted by the system.

Field	Description
bytes output	Total number of bytes, including data and MAC encapsulation, transmitted by the system.
underruns	Number of times that the far-end transmitter has been running faster than the near-end communication server's receiver can handle. This may never be reported on some interfaces.
output errors	Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this might not balance with the sum of the enumerated output errors, as some datagrams might have more than one error, and others may have errors that do not fall into any of the specifically tabulated categories.
collisions	Number of messages retransmitted due to an Ethernet collision. This usually is the result of an overextended LAN (Ethernet or transceiver cable too long, more than two repeaters between stations, or too many cascaded multiport transceivers). Some collisions are normal. However, if your collision rate climbs to around 4-5%, you should consider verifying that there is no faulty equipment on the segment and/or moving some existing stations to a new segment. A packet that collides is counted only once in output packets.
interface resets	Number of times an interface has been reset. The interface may be reset by the administrator or automatically when an internal error occurs.
Restarts	Number of times the controller was restarted because of errors.

Related Commands

A dagger (†) indicates that the command is documented in another chapter.

show interfaces
show ip route †
show *protocol* **route** †

show queuing

To list the current state of the queue lists, use the **show queuing** EXEC command.

show queuing {**custom** | **priority**}

Syntax Description

custom	Shows status of custom queue lists.
priority	Shows status of priority lists.

Command Mode

EXEC

Usage Guidelines

If no keywords are entered, this command show the status of both custom and priority queue lists.

Sample Display

The following is sample output from the **show queuing custom** EXEC command:

```
cs# show queueing custom
Current custom queue configuration:

List   Queue  Args
3      10     default
3      3      interface Tunnel3
3      3      protocol ip
3      3      byte-count 444 limit 3
```

Related Commands

A dagger (†) indicates that the command is documented in another chapter.

custom-queue-list
priority-group
priority-list interface †
priority-list queue-limit †
queue-list default †
queue-list interface †
queue-list protocol †
queue-list queue byte-count †
queue-list queue limit †

show rif

Use the **show rif** command to display the current contents of the RIF cache.

show rif

Syntax Description

This command has no arguments or keywords.

Type

EXEC

Sample Display

The following is a sample display from the **show rif** output.

```
sloth# show rif

Codes: * interface, - static, + remote
Hardware Addr  How   Idle (min)  Routing Information Field
5C02.0001.4322 rg5           -   0630.0053.00B0
5A00.0000.2333 TR0           3   08B0.0101.2201.0FF0
5B01.0000.4444 -             -   -
0000.1403.4800 TR1           0   -
0000.2805.4C00 TR0           *   -
0000.2807.4C00 TR1           *   -
0000.28A8.4800 TR0           0   -
0077.2201.0001 rg5          10   0830.0052.2201.0FF0
```

In the display, entries marked with an asterisk (*) are the router/bridge's interface addresses. Entries marked with a dash (–) are static entries. Entries with a number denote cached entries. If the RIF timeout is set to something other than the default of 15 minutes, the timeout is displayed at the top of the display.

Table 1-21 describes significant fields shown in the display.

Table 1-21 Show RIF Cache Display Field Description

Field	Description
Hardware Addr	Lists the MAC-level addresses.
How	Describes how the RIF has been learned. Possible values include a ring group (rg), or interface (TR).
Idle (min)	Indicates how long, in minutes, since the last response was received directly from this node.
Routing Information Field	Lists the RIF.

shutdown

To disable an interface, use the **shutdown** interface configuration command. To restart a disabled interface, use the **no shutdown** command.

shutdown
no shutdown

Syntax Description

This command has no arguments or keywords.

Default

Enabled

Command Mode

Interface configuration

Usage Guidelines

The **shutdown** command disables all functions on the specified interface. On serial interfaces, this command causes the DTR signal to be dropped. On Token Ring interfaces, this command causes the interface to be deinserted from the ring.

This command also marks the interface as unavailable. To check whether an interface is disabled, use the EXEC command **show interfaces**. An interface that has been shut down is shown as administratively down in the display from this command.

Examples

The following example turns off interface Ethernet 0:

```
interface ethernet 0
shutdown
```

The following example turns the interface back on:

```
interface ethernet 0
no shutdown
```

Related Command

show interfaces

systat

To display information about the active ports of the communication server, enter the **systat** EXEC command.

systat [**all**]

Syntax Description

all (Optional.) Displays information for both active and inactive ports.

Command Mode

EXEC

Example

The following example shows how to use the **systat** command:

```
cs> systat

  Line     User      Host(s)              Idle Location
   0 con 0
   1 tty 1                                      charnel console
   2 tty 2                                      T2500 #1-1
   3 tty 3                                      T2500 #1-2
   4 tty 4   xyz       LANE                  56 T2500 #1-3
   5 tty 5                                      T2500 #1-4
   6 tty 6                                      3262 #A1
   7 tty 7   train   ABC                     0 3262 #B1
   8 tty 8                                      3262 #A2
   9 tty 9   pzwt       XRemote:  6 clients      0 3262 #B2
```

The information displayed includes the line number, connection name, idle time, and terminal location.

transmitter-delay

To specify a minimum dead-time after transmitting a packet, use the **transmitter-delay** interface configuration command.

transmitter-delay *microseconds*
no transmitter-delay

Syntax Description

microseconds	Approximate number of microseconds of minimum delay after transmitting a packet on the MCI and SCI interface cards.

Default

0 microseconds

Command Mode

Interface configuration

Usage Guidelines

This command is especially useful for serial interfaces that can send back-to-back data packets over serial interfaces faster than some hosts can receive them. The **no transmitter-delay** command restores the default.

The transmitter delay feature is implemented for the following Token Ring cards: CSC-R16M, CSC-1R, and CSC-2R. For the first four cards, the command syntax is the same as the existing command and specifies the number of milliseconds to delay between sending frames that are generated by the communication server. Transmitter delay for the CSC-CTR uses the same syntax, but specifies a relative time interval to delay between transmission of all frames.

Example

The following example specifies a delay of 300 microseconds on interface serial 0:

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

tunnel checksum

To enable encapsulator-to-decapsulator checksumming of packets on a tunnel interface, use the **tunnel checksum** interface configuration command. To disable checksumming, use the **no** form of the command.

tunnel checksum
no tunnel checksum

Syntax Description

This command has no arguments or keywords.

Default

no tunnel checksumming

Command Mode

Interface configuration

Usage Guidelines

This command currently applies to generic route encapsulation (GRE) only. Some passenger protocols rely on media checksums to provide data integrity. By default, the tunnel does not guarantee packet integrity. By enabling end-to-end checksums, the communication servers will drop corrupted packets.

Example

In the following example, all protocols will have encapsulator-to-decapsulator checksumming of packets on the tunnel interface:

```
tunnel checksum
```

tunnel destination

To specify a tunnel interface's destination, use the **tunnel destination** interface configuration command. To remove the destination, use the **no** form of this command.

tunnel destination {*hostname* | *ip-address*}
no tunnel destination

Syntax Description

hostname	Name of the host destination.
ip-address	IP address of the host destination.

Default

None

Command Mode

Interface configuration

Usage Guidelines

You cannot have two tunnels using the same encapsulation mode with exactly the same source and destination address. The workaround is to create a loopback interface and source packets off of the loopback interface.

Example

In the following example, the tunnel destination is 131.222.111.234:

```
tunnel destination 131.222.111.234
```

Related Command

tunnel source

tunnel key

To enable an ID key for a tunnel interface, use the **tunnel key** interface configuration command. To remove the ID key, use the **no** form of this command.

tunnel key *key-number*
no tunnel key

Syntax Description

key-number	Integer from 0 to 4294967295.

Default

Disabled

Command Mode

Interface configuration

Usage Guidelines

This command currently applies to generic route encapsulation (GRE) only. Tunnel ID keys can be used as a form of *weak* security to prevent misconfiguration or injection of packets from a foreign source.

Note When using GRE, the ID key is carried in each packet. We do *not* recommend relying on this key for security purposes.

Example

In the following example, the tunnel key is set to 3:

```
tunnel key 3
```

tunnel mode

To set the encapsulation mode for the tunnel interface, use the **tunnel mode** interface configuration command. To set to the default, use the **no** form of the command.

tunnel mode {**cayman** | **eon** | **gre** | **nos**}
no tunnel mode

Syntax Description

cayman	Cayman TunnelTalk AppleTalk encapsulation.
eon	EON compatible CLNS tunnel.
gre	Generic route encapsulation protocol over IP.
nos	KA9Q/NOS compatible IP over IP.

Default

gre

Command Mode

Interface configuration

Example

In the following example, the tunnel mode is set to EON:

```
tunnel mode eon
```

tunnel sequence-datagrams

To configure a tunnel interface to drop datagrams that arrive out of order, use the **tunnel sequence-datagrams** interface configuration command. To disable this function, use the **no** form of this command.

tunnel sequence-datagrams
no tunnel sequence-datagrams

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Interface configuration

Usage Guidelines

This command currently applies to generic route encapsulation (GRE) only. This command is useful when carrying passenger protocols that behave poorly when they receive packets out of order (for example, LLC2-based protocols).

Example

In the following example, the tunnel is configured to drop datagrams that arrive out of order:

```
tunnel sequence-datagrams
```

tunnel source

To set a tunnel interface's source address, use the **tunnel source** interface configuring command. To remove the source address, use the **no** form of the command.

tunnel source {*ip-address* | *interface-type interface-number*}
no tunnel source

Syntax Description

ip address	IP address to use as the source address for packets in the tunnel.
interface-type	All types.
interface-number	Specifies the port, connector, or interface card number. The numbers are assigned at the factory at the time of installation or when added to a system, and can be displayed with the **show interfaces** command.

Default

None

Command Mode

Interface configuration

Usage Guidelines

You cannot have two tunnels using the same encapsulation mode with exactly the same source and destination address. The workaround is to create a loopback interface and source packets off of the loopback interface.

Example

In the following example, the tunnel source is set to the IP address assigned to Ethernet 0:

```
communication server(config-if)# tunnel source ethernet 0
```

Related Command

tunnel destination

tx-queue-limit

To control the number of transmit buffers available to a specified interface on the MCI and SCI cards, use the **tx-queue-limit** interface configuration command.

tx-queue-limit *number*

Syntax Description

number	Maximum number of transmit buffers that the specified interface can subscribe.

Default

Defaults depend on the total transmit buffer pool size and the traffic patterns of all the interfaces on the card. Defaults and specified limits are displayed with the **show controllers mci** EXEC command.

Command Mode

Interface configuration

Usage Guidelines

This command should be used only under the guidance of a technical support representative.

Example

The following example sets the maximum number of transmit buffers on the interface to 5:

```
interface ethernet 0
tx-queue-limit 5
```

Related Command

show controllers mci