Troubleshooting Ethernet

Ethernet was developed by Xerox Corporation’s Palo Alto Research Center (PARC) in the 1970s. Ethernet was the technological basis for the IEEE 802.3 specification, which was initially released in 1980. Shortly thereafter, Digital Equipment Corporation, Intel Corporation, and Xerox Corporation jointly developed and released an Ethernet specification (Version 2.0) that is substantially compatible with IEEE 802.3. Together, Ethernet and IEEE 802.3 currently maintain the greatest market share of any local-area network (LAN) protocol. Today, the term *Ethernet* is often used to refer to all carrier sense multiple access collision detect (CSMA/CD) LANs that generally conform to Ethernet specifications, including IEEE 802.3.

When it was developed, Ethernet was designed to fill the middle ground between long-distance, low-speed networks and specialized, computer-room networks carrying data at high speeds for very limited distances. Ethernet is well suited to applications on which a local communication medium must carry sporadic, occasionally heavy traffic at high peak data rates.

Ethernet and IEEE 802.3

Ethernet and IEEE 802.3 specify similar technologies. Both are CSMA/CD LANs. Stations on a CSMA/CD LAN can access the network at any time. Before sending data, CSMA/CD stations “listen” to the network to see if it is already in use. If it is, the station wanting to transmit waits. If the network is not in use, the station transmits. A collision occurs when two stations listen for network traffic, “hear” none, and transmit simultaneously. In this case, both transmissions are damaged, and the stations must retransmit at some later time. Back-off algorithms determine when the colliding stations retransmit. CSMA/CD stations can detect collisions, so they know when they must retransmit. This access method is used by traditional Ethernet and IEEE 802.3 functions in half-duplex mode. (When Ethernet is operated in full-duplex mode, CSMA/CD is not used.) This means that only one station can transmit at a time over the shared Ethernet.

This access method was conceived to offer shared and fair access to multiple network stations/devices. It allows these systems fair access to the Ethernet network through a process of arbitration by dictating how stations attached to this network can access the shared channel. It allows stations to listen before transmitting and can recover if signals collide. This recovery time interval is called a slot time and is based on the round-trip time that it takes to send a 64-byte frame the maximum length of an Ethernet LAN attached by repeaters. Another name for this shared LAN is a *collision domain*. For half-duplex operation, the mode on which traditional Ethernet is based, the size of your collision domain can be limited by the physical limitations of the cabling utilized. Table 4-1 lists the collision domains for 10/100/1000 Mbps.

Table 4-1 Examples of Traditional Ethernet and IEEE 802.3 Collision Domains

Traditional Ethernet and 802.3 Collision Domains		
Signaling Speed	**Network Diameter**	
10BaseX	About 280 meters (coax)	Ethernet
10/100BaseX	About 205 meters (twisted pair)	IEEE 802.3b
1000BaseX	About 20 meters (fiber and copper)	IEEE 802.3z

The limitations of the cable itself can create even smaller boundaries.

Because the 64-byte slot time is consistent for 10/100/1000 transmission speeds, this severely limits the scalability for 1000BaseX to operate in a network with a diameter of more than 20 meters. To overcome this obstacle, use carrier extension bits in addition to the Ethernet frame size to extend the time that transmits on the wire. This expands the network diameter to 100 meters per segment, like 100BaseT.

For this system to work, everyone must abide by the same rules. For CSMA/CD the rules are as follows:

1. **Listen**—Stations listen for signals on the wire. If a signal is detected (carrier sense), then stations should not attempt to transmit frame. If a station "hears" another signal on the wire while transmitting the first 64 bytes of a frame, it should recognize that its frame has collided with another.
2. **Collision detect**—If a station detects a collision, it must back off from sending the frame using the truncated back-off algorithm. The back-off algorithm counts the number of collisions, if any, to determine how long a station must wait to retransmit the frame. This algorithm backs off each time that a collision is detected. The goal of this method is to provide the system a way to determine how many stations are trying to transmit simultaneously and then guess when it should be safe to try again. The way that the truncated back-off algorithm tracks and adjusts timers is based on the value of 2^n, where n is the number of collisions encountered during transmission of the frame. The result is a guess of how many stations may be on the shared channel. This result gets plugged in as a range, counting from zero, for the number of slot times to wait. The algorithm randomly selects a value from this range as shown in Table 4-2.

Table 4-2 Back-off Algorithm

2^n value[1]	Actions
2^0–1	Stations either try to retransmit immediately or wait for one slot time.
2^2	Stations randomly wait zero, one, two, or three slot times to retransmit.
2^3	Stations randomly wait from zero to seven slot times.
2^4	. . . you get the point.

1. 2^n where n = the number of collisions

Depending on the number of collisions the algorithm randomly selects to back off, a station could potentially wait a while before retransmitting.

The algorithm collision counter stops incrementing at 10, where the penalty wait time is selected from a range of 0 to 1023 slot times before retransmission. This is pretty bad, but the algorithm will attempt to retransmit the frame up to 16 collisions. Then it just gives up, and a higher-layer network protocol such as TCP/IP will attempt to retransmit the packet. This is an indication that you have some serious errors.

When a station successfully sends a frame, the collision counter (penalty) is cleared (for that frame) and no loner must wait for the back-off time. (“Interface” statistics are not cleared, just the timer is). Any stations with the lowest collisions will be capable of accessing the wire more quickly because they do not have to wait.

Both Ethernet and IEEE 802.3 LANs are broadcast networks. In other words, all stations see all frames, regardless of whether they represent an intended destination. Each station must examine received frames to determine whether the station is a destination. If it is a destination, the frame is passed to a higher protocol layer for appropriate processing.

Differences between Ethernet and IEEE 802.3 standards are subtle. Ethernet provides services corresponding to Layers 1 and 2 of the OSI reference model, whereas IEEE 802.3 specifies the physical layer (Layer 1) and the channel-access portion of the link layer (Layer 2), but does not define a logical link control protocol. Both Ethernet and IEEE 802.3 are implemented in hardware. Typically, the physical manifestation of these protocols is either an interface card in a host computer or circuitry on a primary circuit board within a host computer.

Now, having said all that regarding the regular operation of traditional Ethernet and 802.3, we must discuss where the two separate in features and functionality. The IEEE 802.3 standard was based on traditional Ethernet, but improvements have been made to this current standard. What we have discussed so far will not scale in today’s demanding service provider and enterprise networks.

Full-Duplex Operation 10/100/1000

Everything you’ve read so far dealt with half-duplex operation (CSMA/CD, back-off timers, and so on). Full-duplex mode allows stations to transmit and receive data simultaneously. This makes for more efficient use of the available bandwidth by allowing open access to the medium. Conversely, this mode of operation can function only with Ethernet switching hubs or via Ethernet cross-over cables between interfaces capable of full-duplex Ethernet. Full-duplex mode expects links to be point-to-point links. There are also no collisions in full-duplex mode, so CSMA/CD is not needed.

Autonegotiation

Autonegotiation allows Ethernet devices to automatically configure their interfaces for operation. If the network interfaces supported different speeds or different modes of operation, they will attempt to settle on a lower common denominator. A plain repeater cannot support multiple speeds; it knows only how to regenerate signals. Smart hubs employ multiple repeaters and a switch plane internally to allow stations that support different speeds to communicate. The negotiation is performed only when the system initially connects to the hub. If slower systems are attached to the same smart hub, then faster systems will have to be manually configured for 10 Mbps operation.

To make sure that your connection is operating properly, IEEE 802.3 Ethernet employs normal link pulses (NLPs), which are used for verifying link integrity in a 10BaseT system. This signaling gives you the link indication when you attach to the hub and is performed between two directly connected link interfaces (hub-to-station or station-to-station). NLPs are helpful in determining that a link has been established between devices, but they are not a good indicator that your cabling is free of problems.

An extension of NLPs is fast link pulses. These do not perform link tests, but instead are employed in the autonegotiation process to advertise a device's capabilities. Autonegotiation on 1000BaseX networks works at only 1000 Mbps, so the only feature "negotiated" is for full- or half-duplex operation. There may be new vendor implementations on the market that can autonegotiate speeds 10 to 1000BaseX, but at this time they are not widely deployed.

A backup alternative, called parallel detection, works for 10/100 speeds if autonegotiation is disabled or is unsupported. This is basically a fallback mechanism that springs into action when autonegotiation fails. The interface capable of autonegotiation will configure itself for bare bones 10-Mbps half-duplex operation.

Physical Connections

IEEE 802.3 specifies several different physical layers, whereas Ethernet defines only one. Each IEEE 802.3 physical layer protocol has a name that summarizes its characteristics. The coded components of an IEEE 802.3 physical layer name are shown in Figure 4-1.

Figure 4-1 IEEE 802.3 Physical Layer Name Components

A summary of Ethernet Version 2 and IEEE 802.3 characteristics appears in Tables 4-3 and 4-4.

Table 4-3 Ethernet Version 2 and IEEE 802.3 Physical Characteristics

Characteristic	Ethernet Value	IEEE 802.3 Values				
		10Base5	**10Base2**	**1Base5**	**10BaseT**	**10Broad36**
Data rate (Mbps)	10	10	10	1	10	10
Signaling method	Baseband	Baseband	Baseband	Baseband	Baseband	Broadband
Maximum segment length (m)	500	500	185	250	100	1800
Media	50-ohm coax (thick)	50-ohm coax (thick)	50-ohm coax (thin)	Unshielded twisted-pair wire	Unshielded twisted-pair wire	75-ohm coax
Topology	Bus	Bus	Bus	Star	Star	Star

Table 4-4 IEEE 802.3 Physical Characteristics

Characteristic	IEEE 802.3 Values	
	10BaseFX	**1000BaseFX**
Data rate (Mbps)	100	1000
Signaling method	Baseband	Baseband
Maximum segment length (m)	Repeater 150 m; full-duplex 2000 m Single mode up to 6 to 10 km	Repeater 150 m; full-duplex 2000 m Single mode up to 6 to 10 km
Media	Fiber (single mode or multimode)	Fiber (single mode or multimode)
Topology	Star	Star

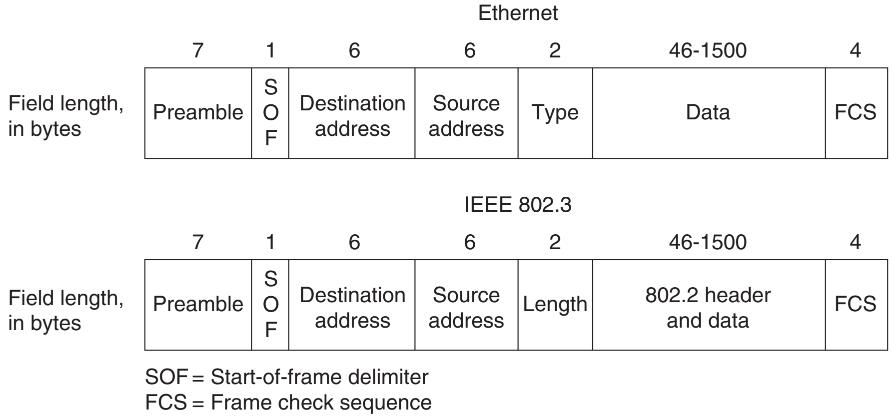
There are other 100Base*n* implementations, but they are not widely implemented for various reasons. One particular case in point is 100BaseT4. This system uses four pairs of copper wire and can be used on voice- and data-grade cable. 10/100BaseT systems perform well on Category 5 data-grade cable and use only two pairs of copper wire.

Ethernet is most similar to IEEE 802.3 10Base5. Both of these protocols specify a bus topology network with a connecting cable between the end stations and the actual network medium. In the case of Ethernet, that cable is called a transceiver cable. The *transceiver cable* connects to a transceiver device attached to the physical network medium. The IEEE 802.3 configuration is much the same, except that the connecting cable is referred to as an *attachment unit interface* (AUI), and the transceiver is called a *media attachment unit* (MAU). In both cases, the connecting cable attaches to an interface board (or interface circuitry) within the end station.

Frame Formats

Ethernet and IEEE 802.3 frame formats are shown in Figure 4-2.

Figure 4-2 Ethernet and IEEE 802.3 Frame Formats



Both Ethernet and IEEE 802.3 frames begin with an alternating pattern of ones and zeros called a *preamble*. The preamble tells receiving stations that a frame is coming.

The byte before the destination address in both an Ethernet and an IEEE 802.3 frame is a start-of-frame (SOF) delimiter. This byte ends with 2 consecutive 1 bits, which serve to synchronize the frame reception portions of all stations on the LAN.

Immediately following the preamble in both Ethernet and IEEE 802.3 LANs are the destination and source address fields. Both Ethernet and IEEE 802.3 addresses are 6 bytes long. Addresses are contained in hardware on the Ethernet and IEEE 802.3 interface cards. The first 3 bytes of the addresses are specified by the IEEE on a vendor-dependent basis, and the last 3 bytes are specified by the Ethernet or IEEE 802.3 vendor. The source address is always a unicast (single node) address, whereas the destination address may be unicast, multicast (group), or broadcast (all nodes).

In Ethernet frames, the 2-byte field following the source address is a type field. This field specifies the upper-layer protocol to receive the data after Ethernet processing is complete.

In IEEE 802.3 frames, the 2-byte field following the source address is a length field, which indicates the number of bytes of data that follow this field and precede the frame check sequence (FCS) field.

Following the type/length field is the actual data contained in the frame. After physical layer and link layer processing is complete, this data will eventually be sent to an upper-layer protocol. In the case of Ethernet, the upper-layer protocol is identified in the type field. In the case of IEEE 802.3, the upper-layer protocol must be defined within the data portion of the frame, if at all. If data in the frame is insufficient to fill the frame to its minimum 64-byte size, padding bytes are inserted to ensure at least a 64-byte frame.

In 802.3 the data field carries a payload header in addition to the payload itself. This header serves the logical link control sublayer of the OSI model and is completely independent of the MAC sublayer and physical layer below it. This header, functionally known as 802.2 encapsulation, contains destination service access point (DSAP) and source service access point (SSAP) information. This will notify higher protocols what type of payload is actually riding in the frame. It functions like the "type" field in traditional Ethernet and is used by upper-layer network protocols such as IPX. Network software developed to support the TCP/IP networking suite uses the type field to determine protocol type in an Ethernet frame. The type field and the LLC header are not replacements for each other, but they serve to offer backward compatibility between network protocol implementations without rewriting the entire Ethernet frame.

After the data field is a 4-byte frame check sequence (FCS) field containing a cyclic redundancy check (CRC) value. The CRC is created by the sending device and is recalculated by the receiving device to check for damage that might have occurred to the frame in transit.

Troubleshooting Ethernet

Table 4-5 provides troubleshooting procedures for common Ethernet media problems.

Table 4-5 Troubleshooting Procedures for Common Ethernet Media Problems

Media Problem	Suggested Actions
Excessive noise	**1.** Use the **show interfaces ethernet** exec command to determine the status of the router's Ethernet interfaces. The presence of many CRC errors but not many collisions is an indication of excessive noise. **2.** Check cables to determine whether any are damaged. **3.** Look for badly spaced taps causing reflections. **4.** If you are using 100BaseTX, make sure you are using Category 5 cabling and not another type, such as Category 3.
Excessive collisions	**1.** Use the **show interfaces ethernet** command to check the rate of collisions. The total number of collisions with respect to the total number of output packets should be around 0.1 percent or less. **2.** Use a TDR to find any unterminated Ethernet cables. **3.** Look for a jabbering transceiver attached to a host. (This might require host-by-host inspection or the use of a protocol analyzer.)
Excessive runt frames	In a shared Ethernet environment, runt frames are almost always caused by collisions. If the collision rate is high, refer to the problem of excessive collisions, earlier in this table. If runt frames occur when collisions are not high or when in a switched Ethernet environment, then they are the result of underruns or bad software on a network interface card. Use a protocol analyzer to try to determine the source address of the runt frames.
Late collisions	**1.** Use a protocol analyzer to check for late collisions. Late collisions should never occur in a properly designed Ethernet network. They usually occur when Ethernet cables are too long or when there are too many repeaters in the network. **2.** Check the diameter of the network, and make sure that it is within specification.
No link integrity on 10BaseT, 100BaseT4, or 100BaseTX	**1.** Make sure that you are not using 100BaseT4 when only two pairs of wire are available. 100BaseT4 requires four pairs. **2.** Check for a 10BaseT, 100BaseT4, or 100BaseTX mismatch (for example, a card different from the port on a hub). **3.** Determine whether there is cross-connect. (For example, be sure that straight-through cables are not being used between a station and the hub.) **4.** Check for excessive noise (see the problem of excessive noise, earlier in this table).

When you're troubleshooting Ethernet media in a Cisco router environment, the **show interfaces ethernet** command provides several key fields of information that can assist with isolating problems. The following section provides a detailed description of the **show interfaces ethernet** command and the information that it provides.

show interfaces ethernet

Use the **show interfaces ethernet privileged** exec command to display information about an Ethernet interface on the router:

- show interfaces ethernet unit [accounting]
- **show interfaces ethernet** [*slot* | *port*] [**accounting**] (for the Cisco 7200 series and Cisco 7500)
- **show interfaces ethernet** [*type slot* | *port-adapter* | *port*] (for ports on VIP cards in the Cisco 7500 series routers)

Syntax Description

unit—This must match a port number on the selected interface.

accounting—(Optional) This displays the number of packets of each protocol type that have been sent through the interface.

slot—Refer to the appropriate hardware manual for slot and port information.

port—Refer to the appropriate hardware manual for slot and port information.

port-adapter—Refer to the appropriate hardware manual for information about port adapter compatibility.

Command Mode

Privileged exec

Usage Guidelines

This command first appeared in Cisco IOS Release 10.0. If you do not provide values for the argument *unit* (or *slot* and *port* on the Cisco 7200 series, or *slot* and *port-adapter* on the Cisco 7500 series), 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:

```
Router# show interfaces ethernet 0
Ethernet 0 is up, line protocol is up
   Hardware is MCI Ethernet, address is aa00.0400.0134 (via 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
            0 input packets with dribble condition detected
        3594664 packets output, 436549843 bytes, 0 underruns
        8 output errors, 1790 collisions, 10 interface resets, 0 restarts
```

Table 4-6 presents **show interfaces ethernet** field descriptions.

Table 4-6 show interfaces ethernet Field Descriptions

Field	Description
Ethernet . . . is up . . . is administratively down	Indicates whether the interface hardware is currently active and whether it has been taken down by an administrator. "Disabled" indicates that the router has received more than 5,000 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 that the interface is usable (that is, whether keepalives are successful) or if it has been taken down by an administrator.
Hardware	Specifies the hardware type (for example, MCI Ethernet, SCI, cBus Ethernet) and address.
Internet address	Specifies the Internet address, followed by the subnet mask.
MTU	Gives the maximum transmission unit of the interface.
BW	Gives the bandwidth of the interface in kilobits per second.
DLY	Gives the delay of the interface in microseconds.
rely	Shows reliability of the interface as a fraction of 255 (255/255 is 100 percent reliability), calculated as an exponential average over 5 minutes.
load	Shows load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over 5 minutes.
Encapsulation	Specifies the encapsulation method assigned to interface.
ARP type	Specifies the type of Address Resolution Protocol assigned.
loopback	Indicates whether loopback is set.
keepalive	Indicates whether keepalives are set.

Table 4-6 show interfaces ethernet Field Descriptions (continued)

Last input	Gives the number of hours, minutes, and seconds since the last packet was successfully received by an interface. This is useful for knowing when a dead interface failed.
Last output	Gives the number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface.
output	Gives the number of hours, minutes, and seconds since the last packet was successfully transmitted by the interface. This is useful for knowing when a dead interface failed.
output hang	Gives the 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	Gives 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 that the elapsed time is too large to be displayed. "0:00:00" indicates that the counters were cleared more than 231ms (and less than 232ms) ago.
Output queue, input queue, drops	Gives the 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	Gives the average number of bits and packets transmitted per second in the past 5 minutes. If the interface is not in promiscuous mode, it senses network traffic it sends and receives (rather than all network traffic). The 5-minute input and output rates should be used only as an approximation of traffic per second during a given 5-minute period. These rates are exponentially weighted averages with a time constant of 5 minutes. A period of four time constants must pass before the average will be within 2 percent of the instantaneous rate of a uniform stream of traffic over that period.
packets input	Gives the total number of error-free packets received by the system. *continues*
bytes input	Gives the total number of bytes, including data and MAC encapsulation, in the error-free packets received by the system.
no buffers	Gives the number of received packets discarded because there was no buffer space in the main system. Compare this with the ignored count. Broadcast storms on Ethernet networks and bursts of noise on serial lines are often responsible for no input buffer events.

Table 4-6 show interfaces ethernet Field Descriptions (continued)

Received . . . broadcasts	Shows the total number of broadcast or multicast packets received by the interface.
Runts	Gives the 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	Gives the number of packets that are discarded because they exceed the medium's maximum packet size. For example, any Ethernet packet that is greater than 1518 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 error 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	Indicates that 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 CRCs is usually the result of collisions or a station transmitting bad data.
frame	Shows the 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	Shows the number of times that the receiver hardware was incapable of handing received data to a hardware buffer because the input rate exceeded the receiver's capability to handle the data.
ignored	Shows the number of received packets ignored by the interface because the interface hardware ran low on internal buffers. These buffers are different from the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be increased.
input packets with dribble condition detected	Gives the dribble bit error, which indicates that a frame is slightly too long. This frame error counter is incremented just for informational purposes; the router accepts the frame.
packets output	Shows the total number of messages transmitted by the system.
bytes	Shows the total number of bytes, including data and MAC encapsulation, transmitted by the system.
underruns	Gives the number of times that the transmitter has been running faster than the router can handle. This may never be reported on some interfaces.

Table 4-6 show interfaces ethernet Field Descriptions (continued)

output errors	Gives the 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 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.
collisions	Gives the 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	Gives the number of times that 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	Gives the number of times a Type 2 Ethernet controller was restarted because of errors.