By Laura Chappell

Murphy's Ghost

An overview of problems that haunt Ethernet and token-ring networks

Many of the most experienced network administrators feel haunted by Murphy's Law as they seek solutions to often-elusive network anomalies. Here are some tips for dealing with common problems on Ethernet and token-ring networks using Novell's LANalyzer for Windows.

Like bumper cars

Ethernet networks are based on the Carrier Sense Multiple Access/Collision Detection (CSMA/CD) scheme. Simply put, this access method requires each workstation to "listen" to the wire before transmitting data. If the wire is busy (there is a signal on the wire), the workstation should defer, or wait, before transmitting onto the network.

If, however, the cable is free, the workstation can transmit a packet. To ensure the packet arrives in good shape, the Ethernet card performs a Cyclic Redundancy Check (CRC) on the packet. This test simply provides a "signature" of the packet contents and enables a receiving workstation to determine if the packet is corrupt.

Transmitting workstations monitor the cable for collisions—a nasty event that occurs when two packets crash (often head-on) on the cable. The result of a collision is fragments on the network. For example, if you drive on the Los Angeles 405 freeway at 5 p.m., most likely, you will see collisions. The cars involved in collisions generally lose a bumper, side-view mirror or perhaps a license plate. Imagine these roadside scraps as fragments on your network. They are clear indications that a collision has occurred.

If a collision is detected, the workstations involved in the collision transmit additional bytes onto the wire. This ensures that all workstations are aware that a collision has occurred. Ethernet workstations can attempt to retransmit onto the network up to 16 times. If they cannot successfully transmit data on the network within 16 attempts, they must give up and notify the protocol that requested transmission (such as NetWare's Internetwork Packet Exchange [IPX]) that the transmission was unsuccessful ("Error Sending on Network XXXX").

Cars involved in collisions generally lose a bumper, side-view mirror or perhaps a license plate. Imagine these roadside scraps as fragments on your network.

Ethernet network errors

A number of hardware-based network analyzers such as Network General's Sniffer hit the market a few years ago, but they were costly and generally difficult to use. A spinoff of these products, Novell's LANalyzer for Windows ($1,495), can be a handy tool for pinpointing and solving many common network problems.

Flaky network interface cards

For example, using LANalyzer for Windows, it's relatively simple to locate a faulty card on an Ethernet network. The program can help you look for workstations that repeatedly send packets with invalid CRC values. Packets that contain an invalid CRC value are considered bad. If these packets consistently are sent from a workstation, it indicates that you should replace the network adapter. In Figure 1, you can see that LANalyzer for Windows has detected CRC errors (as reported on the ticker tape message in the lower left-hand corner) on the network. By sorting LANalyzer for Windows' Station Monitor screen by the "Errors" column, you can isolate the workstation responsible for transmitting faulty packets onto the network. The simple solution: Replace the workstation's network adapter. (See Figure 1.)

Faulty cabling systems

Short circuits are another common Ethernet problem. When cables contain shorts, packets traveling through the faulty section can become corrupt when bits are lost—suddenly the CRC value does not match the packet contents. By sorting LANalyzer for Windows' Station Monitor screen on the errors column, as shown in Figure 2, you can see that CRC errors are distributed among most of the workstations on the network. And, these widespread CRC errors indicate there is a damaged cable on the network. (See Figure 2.)

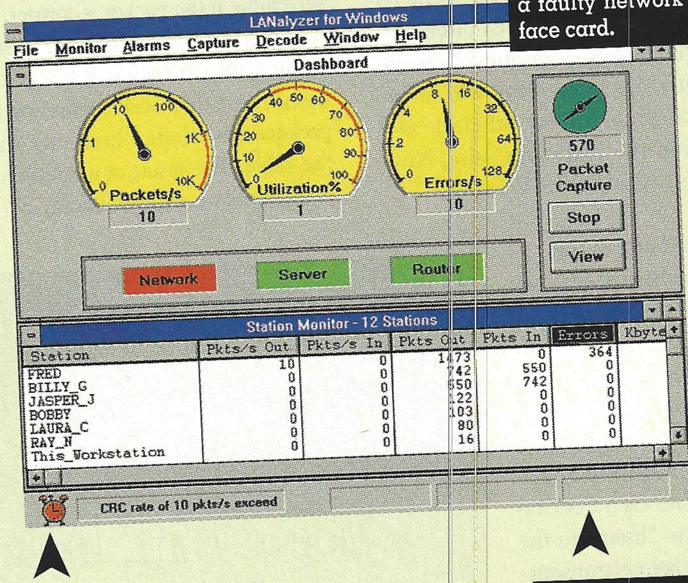
How should a token-ring network work?

Let's take a quick look at a couple of token-ring issues. Often, people who downplay the capabilities of token-ring networks really don't understand how they work.

On a token-ring network, transmitting workstations must wait for a token before they can transmit onto the network. Each workstation is provided equal access to the cabling system. Once a workstation receives a token, it can transmit a frame onto the network. This frame is repeated by each workstation on the ring. Depending on the type of token-ring network, a workstation can either immediately retransmit a token onto the ring (Early Token Release), or wait until its entire frame has been received back, strip off its frame and release a token. In order for a token-ring network to work properly, the ring must be in a closed loop. An open ring cannot support any data. This is the simple part—the complex part of token-ring is the management system.

The token-ring protocol was designed with a robust management system built-in. For example, on a token-ring network, if a network card is suspected of causing an open condition on the ring, it automatically removes itself from the

(Continued on Page 34)

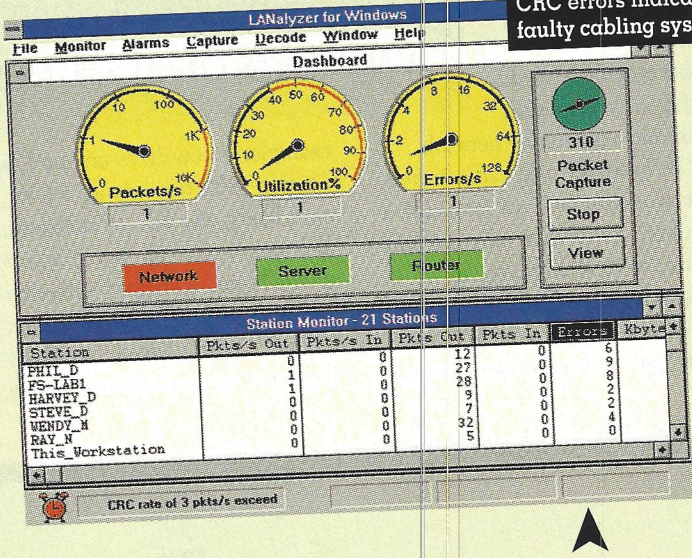


Station Monitor - 12 Stations

Station	Pkts/s Out	Pkts/s In	Pkts Out	Pkts In	Errors	Kbyte
FRED	10	0	1473	0	364	
BILLY_G	0	0	742	550	0	
JASPER_J	0	0	550	742	0	
BOBBY	0	0	122	0	0	
LAURA_C	0	0	103	0	0	
RAY_N	0	0	80	0	0	
This_Workstation	0	0	16	0	0	

CRC rate of 10 pkts/s exceed

Figure 1: Indication of a faulty network interface card.



Station Monitor - 21 Stations

Station	Pkts/s Out	Pkts/s In	Pkts Out	Pkts In	Errors	Kbyte
PHIL_D	0	0	12	0	6	
FS-LAB1	1	0	27	0	9	
HARVEY_D	1	0	28	0	8	
STEVE_D	0	0	9	0	2	
WENDY_M	0	0	7	0	2	
RAY_N	0	0	32	0	4	
This_Workstation	0	0	5	0	0	

CRC rate of 3 pkts/s exceed

Figure 2: Distributed CRC errors indicate a faulty cabling system.

(Continued from Page 32)

ring and tests itself. If it determines that it has an error condition, the card cannot return to the ring. The ring, therefore, is considered to have some self-healing abilities.

Token-ring networks undergo a Ring Poll process every seven seconds. One workstation, designated as the "Active Monitor," announces its presence on the ring. All other workstations on the ring, in turn, transmit a "Standby Monitor Present" packet—a simple announcement that they are on the ring. During this process, each workstation reads the contents of the packets circulating the ring to determine which workstation is their "upstream" neighbor—the workstation that they should expect frames and tokens from. This information is useful when isolating the cause of an open ring, as described below.

The dreaded beaconing ring

What happens if a ring suddenly enters an "open" state? The workstation that first detects the error (the workstation immediately downstream from the open) begins "beaconing"—broadcasting to all workstations on the ring that its upstream neighbor is at fault. On an open token-ring network, no data can be transmitted—only beacon packets are circulating on the ring. Figure 3 shows LANalyzer for Windows indicating that a ring has entered "beaconing." (See Figure 3.)

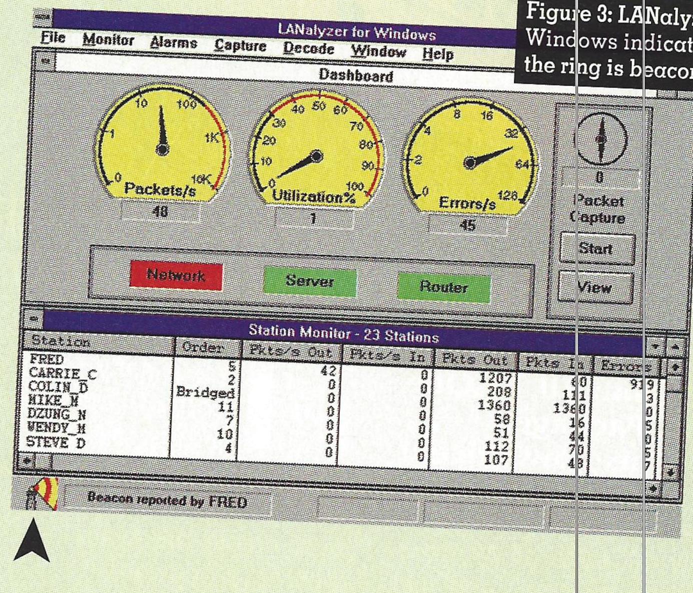
To isolate the fault, you must determine which workstation is upstream from the beaconing workstation. This indicates the area of the fault. Check the cabling of the multistation access unit (MAU) to ensure that Ring In and Ring Out ports are empty unless properly wired to another MAU. If the wiring looks acceptable, unplug and replug the beaconing workstation from the MAU. Then unplug and replug the beaconing workstation's upstream neighbor from the MAU. This is a simple method for checking the relay mechanism in the MAU. If the beaconing condition is cleared immediately after disconnecting one of the workstations, one of your ports on the MAU is faulty.

The overloaded token-ring card

Another common problem occurs when a token-ring card cannot keep up with network traffic. Normally, when a packet is addressed to a card on a token-ring network, four bits in the packet indicate if the address was recognized and if the packet was copied into the receiving workstation's buffer. If the card is overloaded, it sets only two of the bits in the packet—indicating that it recognizes its address, but cannot copy the packet into its buffer because its buffer is "full." This condition is called "Receiver Congestion." Figure 4 shows the Ring Order screen of LANalyzer for Windows. This screen displays the statistics of each ring workstation. (See Figure 4.)

While network analysis used to be the domain of diehard techies, LANalyzer for Windows and similar packages promise to free network administrators from some of the hassles and costs of seeking expert assistance. NS

Laura Chappell of ImagiTech specializes in networking issues. You can reach her at 408-321-1211.

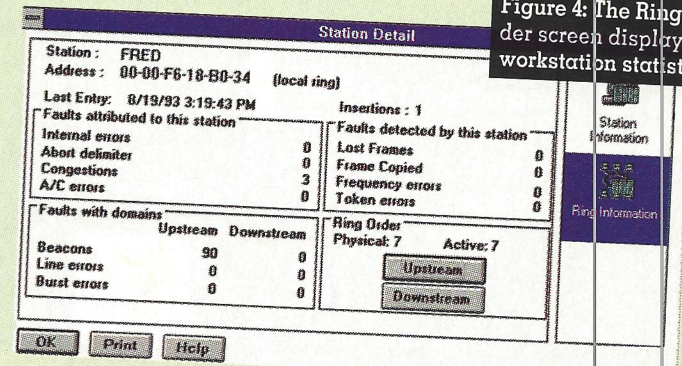


Station Monitor - 23 Stations

Station	Order	Pkts/s Out	Pkts/s In	Pkts Out	Pkts In	Errors
FRED	5	42	0	1207	60	919
CARRIE_C	2	0	0	208	111	3
COLIN_D	Bridged	0	0	1360	1360	0
MIKE_M	11	0	0	58	16	5
DZUNG_N	7	0	0	51	44	0
WENDY_M	10	0	0	112	70	5
STEVE D	4	0	0	107	48	7

Beacon reported by FRED

Figure 3: LANalyzer for Windows indicates the ring is beaconing.



Station Detail

Station : FRED
Address : 00-00-F6-18-B0-34 (local ring)
Last Entry: 8/19/93 3:19:43 PM
Insertions : 1

Faults attributed to this station

Internal errors	0
Abort delimiter	0
Congestions	3
A/C errors	0

Faults detected by this station

Lost Frames	0
Frame Copied	0
Frequency errors	0
Token errors	0

Faults with domains

	Upstream	Downstream
Beacons	90	0
Line errors	0	0
Burst errors	0	0

Ring Order
Physical: 7 Active: 7

Figure 4: The Ring Order screen displays ring workstation statistics

Congestion Relief

Regardless of whether you're operating an Ethernet or token-ring LAN, a growing network is bound to experience congestion problems at some point in time. NetWare has a congestion control mechanism that lets the file server inform clients that it cannot process a request at a given time. As more NetWare Loadable Modules (NLMs) and applications are loaded on a NetWare file server, this congestion control mechanism indicates the amount of work a server can perform optimally.

Let's examine how the communications between a NetWare client and file server should work, then how to identify an overloaded server and relieve congestion.

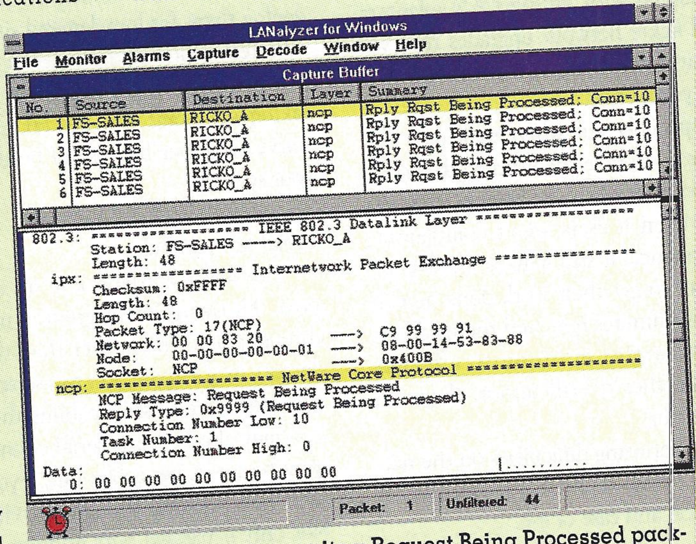
In a perfect world...

There are two primary NetWare Core Protocol (NCP) types used for most NetWare communications—NCP Request (type 2222) and NCP Reply (type 3333). NetWare clients that desire file, print or other services transmit an NCP Request to the file server. The file server, in turn, processes the request and sends an NCP Reply.

For example, to load an application that is stored on the server drive a client issues an NCP Request to open the file. The server transmits an NCP Reply indicating the request was executed successfully if the user has sufficient rights to open the file and the file was found in the user's drive mappings. The response from an overloaded file server looks much different.

"Just a moment, please"

If the NetWare server cannot process the client's request because it is busy, the server transmits a special packet type 9999, "Request Being Processed." This packet indicates to the client that the server received the request successfully, but the server is asking the client to wait for a response until the request can be processed. Upon receiving this Request Being Processed packet, the client resets its IPX time-out counter and waits. If it doesn't receive a response within the time-out period, the client re-executes the request.



```
No. Source     Destination  Layer  Summary
  1 FS-SALES   RICKO_A      ncp    Rply Rqst Being Processed; Conn=10
  2 FS-SALES   RICKO_A      ncp    Rply Rqst Being Processed; Conn=10
  3 FS-SALES   RICKO_A      ncp    Rply Rqst Being Processed; Conn=10
  4 FS-SALES   RICKO_A      ncp    Rply Rqst Being Processed; Conn=10
  5 FS-SALES   RICKO_A      ncp    Rply Rqst Being Processed; Conn=10
  6 FS-SALES   RICKO_A      ncp    Rply Rqst Being Processed; Conn=10

802.3: =================== IEEE 802.3 Datalink Layer ===================
       Station: FS-SALES ----> RICKO_A
       Length: 48
  ipx: ================= Internetwork Packet Exchange ================
       Checksum: 0xFFFF
       Length: 48
       Hop Count:  0
       Packet Type: 17(NCP)
       Network: 00 00 83 20          ---->  C9 99 99 91
       Node:    00-00-00-00-00-01    ---->  08-00-14-53-83-88
       Socket:  NCP                  ---->  0x400B
  ncp: ===================== NetWare Core Protocol =====================
       NCP Message: Request Being Processed
       Reply Type: 0x9999 (Request Being Processed)
       Connection Number Low: 10
       Task Number: 1
       Connection Number High: 0
Data:
   0: 00 00 00 00 00 00 00 00 00 00                   |..........
```

Figure 1: The server transmits a Request Being Processed packet when it cannot service a request.

Figure 1 shows the structure of a Request Being Processed packet.

What creates an overloaded server?

Several factors can cause an overloaded server—and each requires some output of cash to fix. Using LANalyzer for Windows, you can track the number of Request Being Processed packets as they begin to rise. This is the time to begin filling out that purchase requisition.

First, if you are one of the many folks who are running NetWare v3.x or 4.x on a 386SX CPU, upgrade the processor. Most likely, you will be happier with the performance of a full 386 or 486 chip—it will handle more requests and reduce the number of times the server says, "Please wait."

Second, check the amount of memory in your server. If you are scraping by with the minimum amount of memory, chances are you don't have enough file cache buffers. This means that the server must go to disk for many of the file read requests—a relatively slow process compared to RAM-serviced requests. When the server is constantly busy going to disk, it cannot process some of the other requests coming into it.

Third, look at the responsibilities you have placed on the server. Is this server supporting numerous I/O-intensive NetWare Loadable Modules (NLMs)? If so, the server could be spending much of its time servicing the requests of your database engine NLMs, performance tracking NLMs or management NLMs. Not much OS power is left for those crucial reads and writes.

Consider load balancing between servers on the network. To load balance, move the NLMs to another server that can support some of the NLMs that currently reside on the overloaded server. Unfortunately, this can be the most expensive solution if you only have a single server on your network—you must purchase an additional server and operating system copy.

As NetWare LANs increase in size and complexity, the likelihood of experiencing the "Overloaded Server Syndrome" also increases. Keeping a close watch on these Request Being Processed packets can assure you ample time to plan your upgrade.

—Laura Chappell