NETWORKING NETWORKS: Integrating Other LANs To NetWare

LAN and Multiuser PC Integration

October 1988 • $3.95 USA (Canada $4.95)

Micro Systems JOURNAL

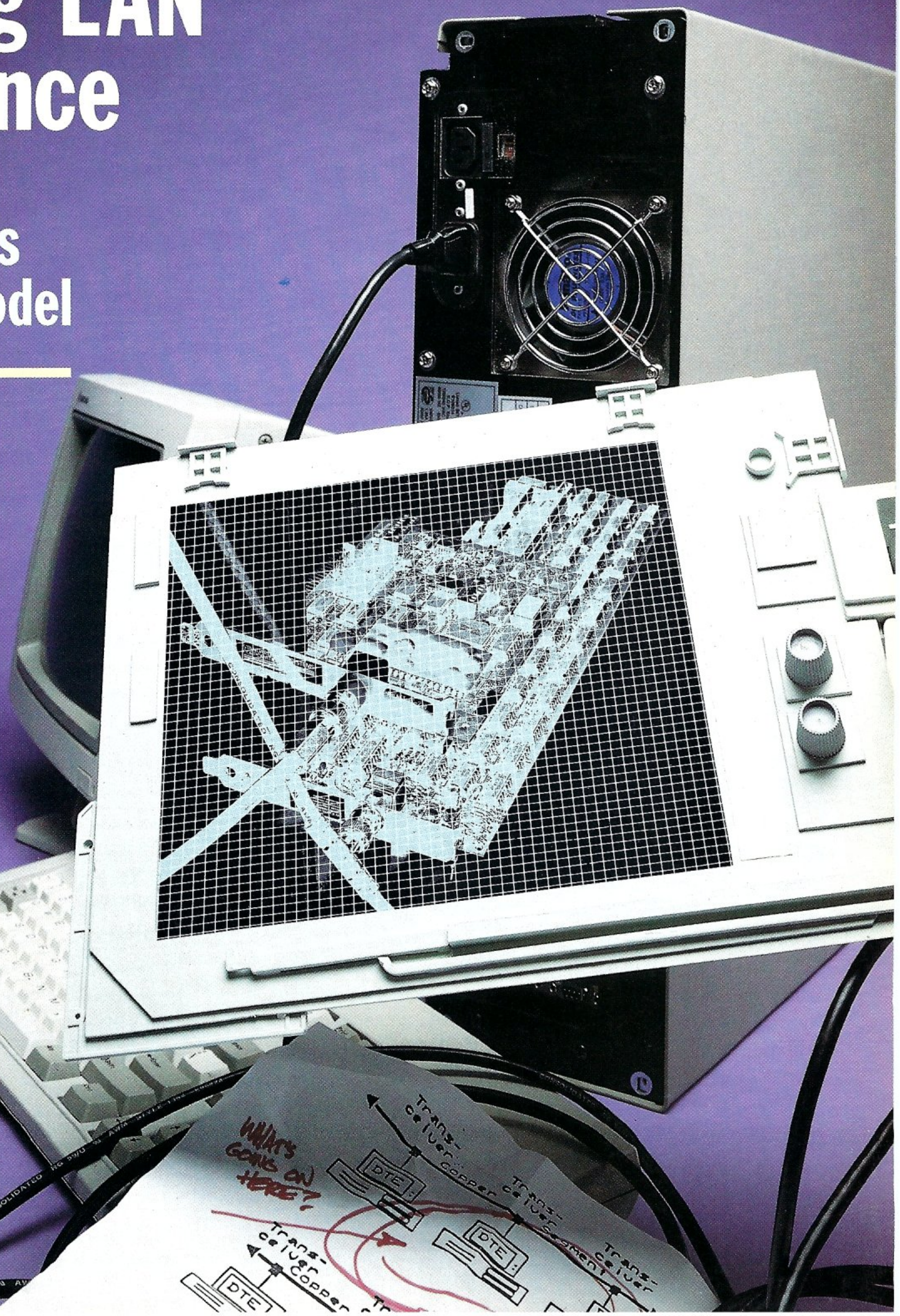
Examining LAN Performance

Diagnosing LANs with the OSI Model

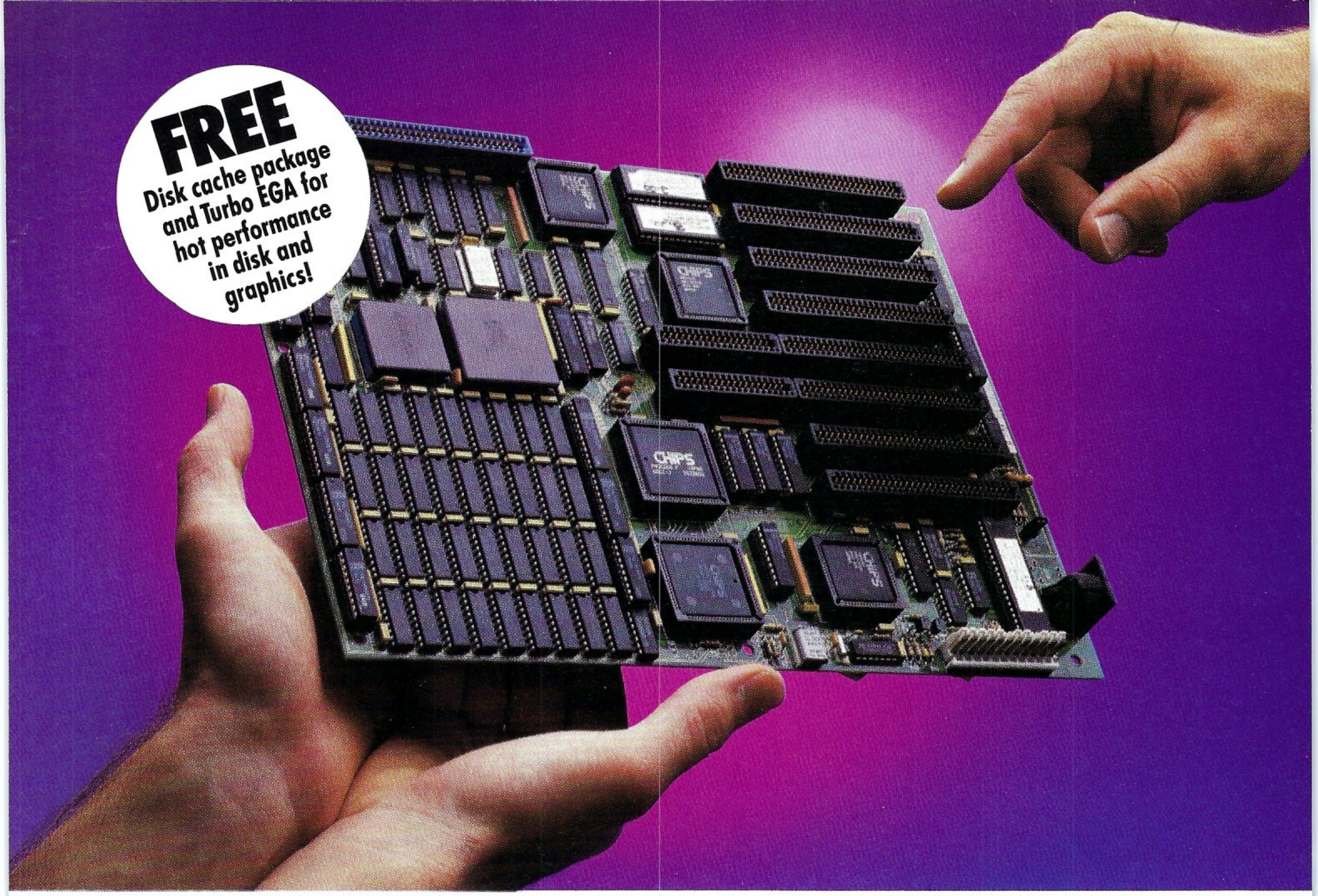
A Benchmark for Your Network

LAN Analyzers In Action

Comparing ARCNET Boards



10
0 38351 19902 9



CONVERT YOUR PC, XT OR AT INTO A HIGHER FORM OF LIFE!

386 MOTHERBOARDS FOR 386 SPEED

Don't let your PC give up the ghost—Hauppauge has just arrived with a new spark of life: the 386 MotherBoard.™ Far more advanced than an accelerator card, our line of MotherBoards grace your PC, PC/XT, PC/AT or compatible with speeds equal to the IBM PS2 Model 80. And faster. Because we've built in 1 Megabyte of high speed RAM and a 387 math coprocessor socket for speeds that make users humble with awe.

OS/2 Compatible. To ensure a long, fruitful life, our 386 MotherBoard is compatible with the PC/AT (BIOS and I/O)—so you can run the new generation of DOS, OS/2. Our Board also runs Windows/386, UNIX V and PC-MOS/386. For more power, you'll find 16-bit expansion slots that accommodate the latst I/O expansion cards. No 386 accelerator card gives you so much versatility. **Only our 386 MotherBoard gives your PC a future with unlimited possibilities!**

The Critics Applaude! *PC Magazine* awarded our Board "The Editor's Choice" for 386 Replacement Boards. *PC World* called it "the Upgrade Product of the Year."

Technical Features ▪ 16 MHZ 80386 ▪ 1 Megabyte of 100 nsec 4-way interleaved RAM ▪ PC/AT compatible I/O and BIOS for support of OS/2 ▪ Six 8-bit expansion slots ▪ two 16-bit expansion slots (four on 386 MotherBoard/AT) ▪ One 32-bit expansion slot for up to 12 Megabytes of high speed memory ▪ Battery-powered clock calendar

386 MotherBoard/PC or MotherBoard/XT	$1495
386 MotherBoard/AT	$1595
32-bit RAM Board (2 Mbytes installed; up to 4 Mbytes)	$795
16 MHz 80387 math coprocessor	$695
16-bit combination hard disk/ floppy disk controller	$245

For more information on our easy-to-install MotherBoard, call: **1 (800) 443-6284**. In New York, call **(516) 434-1600**.

Hauppauge Computer Works, Inc.
175 Commerce Drive,
Hauppauge, New York 11788

Hauppauge!

CIRCLE 105 ON READER SERVICE CARD

Our new LAN Catalog is guaranteed to knock you off your feet.

You're a LAN manager, and you need answers to networking questions. So you hunt through manuals and files; call service people on both coasts; run programs and just plain run around...

Now, you can slow down, take time to do the job better, and maybe get home before dark. You have somewhere to turn for help: *The LAN Catalog* from **Black Box.**

The LAN Catalog is a single reference guide featuring products and information on all major LAN technologies — Ethernet, IBM® Token Ring, ARCNET®, Starlan and more. Included in the catalog are products for NOVELL®, 3Com®, AppleTalk®, DECNET™, Banyan®, TCP/IP and other NOS environments.

At **Black Box**, we know products alone don't solve problems. You need a single source of information to help you choose and install LAN products. And down the road, you need someone to answer questions about upgrade paths and add-on options. *The LAN Catalog* is much more than products — it's the promise of **FREE, expert Technical Support.** (Technical Support has been the ***Black Box Difference*** for more than 11 years.) Our LAN Technical Support Team is available by phone to answer your networking questions.

Call (412) 746-5151 to receive your free copy of the premiere issue of *The LAN Catalog* from Black Box.

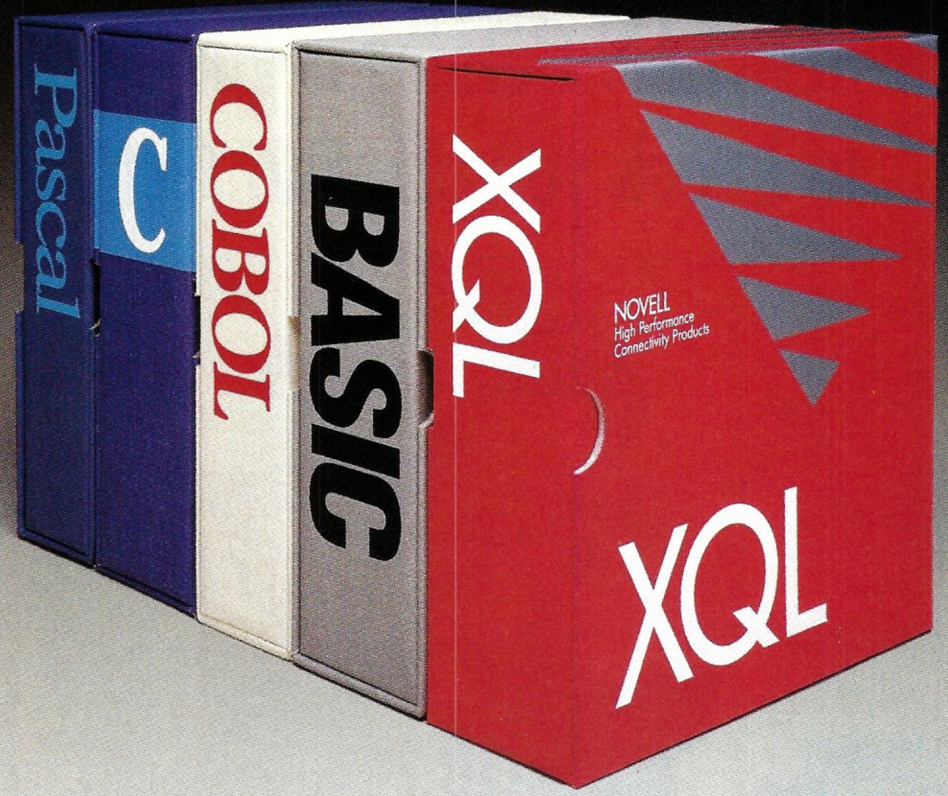
To receive your *free* copy of the premiere issue of *The LAN Catalog* from **Black Box**, complete the coupon and mail to:

Black Box Corporation
P.O. Box 12800
Pittsburgh, PA 15241

Name ______________________________
Title ______________________________
Company ______________________________
P.O. Box ______________________________
Address ______________________________
City __________ State __________ Zip __________
Phone ( ) ______________________________

Allow four weeks for delivery. To receive your catalog in just three days, call (412) 746-5151.

CIRCLE 119 ON READER SERVICE CARD.



Finally, there's an SQL that gets back to BASIC. And COBOL. And C. And Pascal.

As a programmer, you've probably already faced it—the database dilemma. Do you use an SQL for easy database handling, or a true programming language for maximum power and flexibility?

Now you can do both with XQL,® the relational data management system from the developers of Btrieve.®

The Programmer's SQL. With XQL, you can access your data with the ease of Structured Query Language through simple subroutine calls from traditional programming languages. XQL supports standard SQL syntax, including subqueries, unions and security groups.

XQL Relational Primitive Operations. In addition, XQL lets you bypass the SQL level and perform highly efficient, relational primitive operations directly. You get all the functionality of a relational database model without the constraints of a 4th generation language.

Building on Btrieve. The heart of Novell's family of data management tools is Btrieve. By letting you access multiple records at a time, XQL adds a powerful dimension to Btrieve. XQL incorporates sophisticated data manipulation features which allow you to access data by field name, move forward or backwards through the database, compute fields from other fields or constants, and even work with composite records built from multiple, joined Btrieve files.

Like Btrieve, XQL offers features like multi-user support, fault tolerance, comprehensive documentation, and expert technical support. And you never pay royalties on your XQL applications.

Solve the database dilemma with XQL, the SQL that speaks your language. Only $795.* See your Authorized Novell Gold Reseller, or call us at (512) 346-8380.

For more information, call from your modem 1-800-444-4472 (8 bit, no parity, 1 stop bit) and enter the access code NVXQL13.

NOVELL

For software solutions, you should be seeing red.

*Suggested retail price (US dollars) ©1988 Novell Inc., World Headquarters, 122 East 1700 South, Provo, Utah 84601 (801) 379-5900
Requires Btrieve 4.x and PC-DOS or MS-DOS 2.x, 3.x.

Circle Reader Service Number 65

LAN and Multiuser PC Integration
Micro/Systems JOURNAL

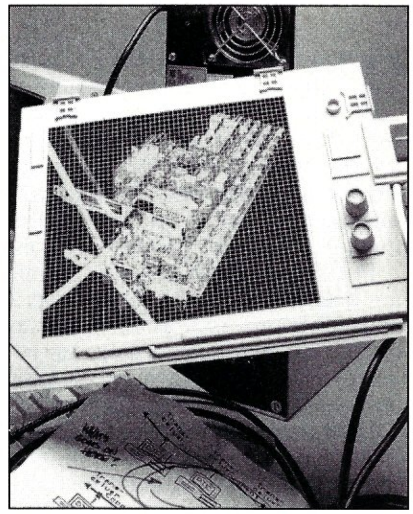
October 1988
Vol. 4 . No. 10

FEATURE ARTICLES

PRODUCT PROFILES

PRODUCT REVIEWS

COLUMNS



About the cover: Like the doctor, the network diagnostician has to isolate clogged coaxial arteries and make sure that the file server heart of the network is pumping data to the workstation organs. And the LAN doctor has a variety of diagnostic tools at his disposal. The question is, what tools are available and how do you determine which one to use? In this issue, *Micro/Systems* focuses on LAN analysis, discussing both the theory behind isolating LAN faults and offering a closer look at some devices that will make the LAN doctor's task an easier one.

DEPARTMENTS

Micro/Systems Journal (ISSN 8750-9482) is published monthly by M & T Publishing, Inc., 501 Galveston Drive, Redwood City, CA 94063; (415) 366-3600. Second-class postage paid at Redwood City and at additional entry points. **POSTMASTER:** send address changes (form 3547) to *Micro/Systems Journal*, Box 3713, Escondido, CA 92025. **Change of Address:** Please send old label and new address to *Micro/Systems Journal*, Box 3713, Escondido, CA 92025. **Customer Service:** For subscription orders and changes of address, call toll-free: (800) 321-3333; in California call (800) 331-4164; outside U.S. call (619) 485-9623. **Subscription Rates:** U.S., $29.97 for one year, $56.97 for two years. Foreign orders must be prepaid in U.S. funds with additional postage. Add $10 per year for surface mail to all countries or add $20 for airmail to Canada and Mexico; $28 for airlift/surface to other countries.

LAN Analysis: More Than Just Stopping the Leaks

The advent of local area networks has increased the power available to the PC user. Sharing applications and data gives accountants, scientists, developers, marketers, and even magazine editors more power at the workstation level, which increases individual productivity and creativity. LANs are the conduits, the "plumbing" if you will, between the file server and the workstation, between the data and the spreadsheet. Your job as network manager is to keep the network running at peak efficiency, or to extend the metaphor, to keep the pipes free of clogs so the data can flow freely.

But the tangle of network plumbing is becoming more complicated. LANs are taking on more and more operating layers as more and more devices are added; various network protocols are interconnected; PCs, micros, and mainframes are linked; and more users are jockeying for access to the same file servers. And as the network plumbing becomes more complicated, LAN analysis devices become a critical part of the system integrator's toolbox.

Your first responsibility is to keep the network up and running smoothly. There is nothing more frustrating than to be working on a complicated project only to have disruption in the network flow propel your hard work into LAN limbo. (In fact, you are reading the second draft of this editorial. The first draft fell victim to a file server crash.) Of course, we all know how truly challenging it is to keep things running smoothly. And when things do go wrong, you have to know how to find the fault and correct it quickly.

The beginnings of network maintenance were somewhat like repairing the first automobile air conditioners; the only way to fix it was to keep changing parts until you found the part that was causing the problem. Some engineers are still using this approach. Consider the case of the failed ARCNET network presented by Paul Cosgrove in this issue. The only solution the hapless engineer-in-residence could apply was to swap active hubs to improve system performance. But as Paul demonstrates, there is no need to stumble around in the dark. With the help of LAN analysis software, he found the real culprit, a faulty ARCNET board, and restored network performance almost immediately.

Dedicated LAN analysis hardware and software make it easy to spot leaks in the network plumbing. Look at the benefits the Stanford Medical School has gained from protocol analysis. In this issue, Phil Fernandez describes how he keeps 400 users on eight Ethernet systems connected and productive with the aid of a protocol analyzer. Such diagnostic tools can prove invaluable in isolating all sorts of network bottlenecks —faulty network cards, bad cabling, improper network protocols, all kinds of software and hardware failures. Graphically plotting network traffic makes it easier to silence the screams of frustrated network users in short order.

There is another benefit to LAN analysis that few consider, benchmarking. Using benchmarking programs, such as the home-made LAN barometer offered by Scott Taylor, provides a measurement of performance that can facilitate network expansion. Monitoring the rate at which information flows through the network provides an early warning mechanism to alert you when new network software or the addition of new users and devices is adversely affecting performance. Demands on the network are certain to increase. LAN-to-LAN communications are becoming widespread. UNIX-based multiuser hosts are being added to existing network environments. And what about the demands that multitasking with OS/2's LAN Manager will place on network resources? The only practical method to monitor the impact of these additional stresses is benchmarking.

The articles in this issue of *Micro/Systems* are presented to help you select new analysis tools for your toolbox. They may even help you develop some tools of your own! So the next time the phone starts ringing and someone starts screaming about network failure, don't reach for the aspirin bottle, reach for your LAN analyzer instead.

Thos M Woolf

Thomas M. Woolf
Managing Editor

LAN and Multiuser PC Integration
Micro/Systems JOURNAL

EDITORIAL
Founder and Editor Sol Libes
Technical Editors Stephen R. Davis, Don Libes
Associate Editors Lennie Libes, Susan Libes
Contributing Editors A.G.W. Cameron, Patrick Corrigan, P.L. Olympia
Managing Editor Thomas M. Woolf

PRODUCTION
Art & Production Director Larry L. Clay
Art Director Joe Sikoryak
Asst. Art Director Barbara Mautz
Typographers Lorraine Buckland, Margaret Anderson

CIRCULATION
Director of Circulation Maureen Kaminski
Asst. Circulation Mgr. Andrea Weingart
Fulfillment Coord. Francesca Martin
Newsstand Mgr. Sarah Frisbie

ADMINISTRATION
V.P. Finance & Operations Kate Wheat
Business Mgr. Betty Trickett
Accts. Payable Asst. Luanne Rocklewitz
Accts. Receivable Asst. Wendy Ho

ADVERTISING
National Account Mgr. Dwight Schwab
National Account Mgr. Tami Brenton
Advertising Coord. Shaun Hooper

Associate Publisher Michael Wiener

M&T Publishing, Inc.

Chairman of the Board Otmar Weber
Director C.F. Von Quadt
President & Publisher Laird Foshay
V.P. of Publishing William P. Howard

Micro/Systems Journal (ISSN 8750-9482) is published monthly by M&T Publishing, Inc., 501 Galveston Drive, Redwood City, CA 94063; (415) 366-3600. Second-class postage paid at Redwood City and at additional entry points.

Article Submission: If you have a specific area of expertise or interest and would like to contribute, please write Micro/Systems Journal, P.O. Box 1192, Mountainside, NJ 07092; (201) 522-9347, or contact M&T Publishing, Inc., 501 Galveston Drive, Redwood City, CA 94063; (415) 366-3600. Please do not submit articles without first contacting the editors.

Correspondence: Please send letters to the editor to Micro/Systems Journal, 501 Galveston Drive, Redwood City, CA 94063. Other editorial correspondence may also be directed to P.O. Box 1192, Mountainside, NJ 07092. The editors may also be reached via MCI Mail (SLIBES or MSJ).

Advertising Rates: Available upon request. Call (415) 366-3600 or write to: Advertising Department, Micro/Systems Journal, 501 Galveston Drive, Redwood City, CA 94063.

Back Issues: Available through M&T Books, 501 Galveston Drive, Redwood City, CA 94063 or by calling toll-free 800-533-4372 (in CA 800-356-2002). Price: 1 issue $5.00, 2-5 issues $4.50 each, 6 or more issues $4.00 each.

News & Views

by Sol Libes

Random Rumors & Gossip

IBM has indicated that it will soon upgrade its PS/2 machines with super-VGA graphics display systems based on the 8154/A controller chip. The displays will offer 1,024-by-728 pixel resolution in up to 256 colors from a palette of 262,144 colors.

Apple Computer has released Version 6.0 of its Macintosh operating system software. That's twice as many versions as PC/MS-DOS released in almost half the time.

The United States Department of Commerce has released figures projecting U.S. computer shipments of $66 billion for this year compared to $56 billion each in 1987 and 1986, and $62 billion in 1985.

Intel, which will begin shipping an 80386 chip rated for a 33-MHz clock rate early next year, is now promising to release a 40-MHz chip version in 1990.

Advanced Graphics Applications Inc. (AGA) of New York, N.Y., has shown prototypes of rewritable optical disk subsystems for PC/AT systems. The $5^1/_4$-inch drive stores 650 MB, includes DOS-OS/2 drivers, and will sell for $4,995. The rewritable optical subsystem is scheduled for shipment in November.

Compaq Computer reports that OS/2 is now being shipped with 5 percent of its Deskpro 286s systems.

3Com Corporation is boasting that it will be the first company to ship an OS/2 LAN Manager-based product. 3Com is scheduled to begin shipping its 3+Open Network operating system this month.

Toshiba Corporation has introduced a $3^1/_2$-inch floppy drive with a maximum storage capacity of 4 megabytes.

IBM has announced that it shortly expects to upgrade the RT PC RISC-based system from 4.5 mips to 6 mips.

286 Systems Will Dominate

Dataquest, a market research company, reports that sales of 80286 systems in 1989 will exceed 6.5 million units. This will surpass 8088/8086-based systems, which are expected to reach 4.9 million systems in 1989. Dataquest reports that sales of 8088/8086 systems are declining and company executives predict that sales will drop under 2 million by 1992 as the 80286-based systems become the low-end machines, at which time they expect 80386 systems shipments to overtake 80286 shipments with combined sales of 19.6 million units.

IBM Ships PC-DOS 4.0

IBM finally shipped PC-DOS Version 4.0 in August. As rumored in this column well over six months ago, it includes a new graphical user interface, built-in file manager, support for extended memory (using the IBM XMA extended memory adapter), and raises the hard-disk partition limit to 512 MB.

This new DOS version was developed by IBM and will be licensed to Microsoft for relicensing to OEMs. This should present an interesting problem for Microsoft which had previously introduced MS-DOS Version 4.0 with multitasking features to several OEMs outside of the United States.

It should also be noted that the features of PC-DOS 4.0 have been available for some time via DOS add-on products. Further, users running DOS on 80386 systems have other add-on products available to them to increase the power of DOS far beyond the features added by IBM in this new release. Specifically, I am referring to DOS extender products from Quarterdeck Office Systems (Santa Monica, Calif.) and Phar Lap Software (Cambridge, Mass.) that allow multitasking of many standard DOS applications.

OS/2 Database Applications Begin Shipping

The first products to take advantage of OS/2's multitasking capabilities are several new OS/2 versions of popular database managers. Paradox OS/2 (Borland International), R:base for OS/2 (Microrim), Informix 4GL (Informix Software), Dataflex OS/2 (Data Access), and Oracle OS/2 (Oracle Corporation) all now feature multitasking multiple sessions and thereby claim significant performance benefits.

IBM Ships OS/2 EE

In July, IBM began shipping the OS/2 Extended Edition (EE), only 16 months after announcing the product. The operating system is intended to connect PS/2 machines to IBM mainframe host systems, however, it lacks the key communications features that IBM promises will be released soon (no availability dates have been announced as yet). The Presentation Manager version, which will provide a graphic windowing interface, is promised for shipment in November. The operating system, which requires a minimum of 6 MB of RAM, retails for $795.

Microsoft is promising that although it will not have an Extended Edition of its OS/2 implementation, it will add EE functionality to its OS/2 package.

Atari Delays Workstations

Atari Corporation, which announced at last November's Comdex that it would introduce a workstation based on Inmos Corporation's T-800 Transputer parallel processing chip set, has delayed introduction of the product until early next year. Although the hardware is ready to be shipped, the operating system is not.

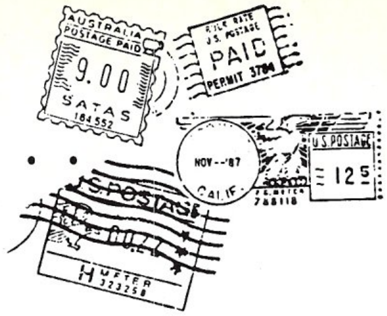
Atari also had demoed a prototype of a powerful low-cost UNIX-based 68030-based workstation at the Hanover fair in the spring, promising fall delivery. Atari now says the system is being redesigned to keep the cost down and also that it will not be available until next year.

Jack Tramiel, CEO of Atari, has a long history of announcing products and showing prototypes and not going into actual production until the company begins to receive orders. It's a neat way to do business, and to throw your competition off balance.

Quotation of the Month

When asked about sales of the PS/2-compatible systems, Ed Juge, Director of Market Planning for Tandy Corporation, replied, "There's not an earth-shattering demand for those products." □

there is mail . . .



Civilizing Assembly Primitives

Dear Editor,

I would like to point out an inaccuracy in Thomas Dwyer's article, "Assembly Language Primitives for EGA Graphics Under Turbo C" (*Micro/Systems*, July 1988), and suggest a modification to his dot plotting algorithm that results in a significant speed increase. First, the inaccuracy has to deal with his statement that "the memory is called VRAM (Video RAM)."

The memory on the EGA card is not VRAM. Most memory technology standards today recognize a VRAM chip as a Multi-Port DRAM. This distinction is very important to vendors such as Video Seven that use VRAM technology to improve the throughput of its state-of-the-art VGA boards. The concept behind the Multi-Port Video RAM (VRAM) eliminates the large bottleneck caused by dedicating a high-speed serial port to the refresh function.

I'm quite certain that Professor Dwyer and his students were aware of this subtle distinction. I believe his use of the acronym VRAM was more for brevity than anything else.

Now for my modification to his dot plotting routine. Professor Dwyer did an excellent job of stressing the importance of optimizing the code to maximize the throughput. Unfortunately, he missed a significant opportunity to improve upon the address calculation routing, code that occupies approximately 25 percent of execution time for version 2 and 3 of the EGADOT procedure.

His solution is based upon the premise that a sequence of left shifts and adds are faster than corresponding multiplication by 80. He is, of course, correct. However, application of the familiar space-time trade off produces even better results. (The space-time trade off basically says that program code space can usually be traded at the expense of execution time.) This is exactly what my solution proposes.

The idea is simple. Let's move all multiplications by 80 or sequences of left shifts and adds from program execution time to program assembly time. This is done through the use

Listing 1.

```
;-----------------------------------------------------------------------
;PLOTDOTH -- Ultra fast version of hi res dot plotting on CGA Adapter
;    Completely table driven.
;
;Author :  Bob Zigon
;Date   :  June 23, 1988
;-----------------------------------------------------------------------

;-----------------------------------------------------------------------
;                             Global Data
;-----------------------------------------------------------------------
dseg      segment word public 'DATA'

;
;-------- lookup table for start of each graphics line
;-------- index is: (Y * 2)
;
row_tbl   label word
  addr = 0
  rept 100
     dw addr,addr+2000H    ;Y=0,1; 2,3; 4,5; etc
     addr = addr+80
  endm
;
;-------- lookup table for hi-res pixel positions in relevant byte
;-------- index is: (X mod 8)
;
mask_tbl  db 01111111b, 10111111b, 11011111b, 11101111b
          db 11110111b, 11111011b, 11111101b, 11111110b


;
;-------- lookup table for color, according to position in byte
;-------- index is (COLOR * 8) + (X mod 8)
;
color_tbl db 00000000b, 00000000b, 00000000b, 00000000b    ;color 0
          db 00000000b, 00000000b, 00000000b, 00000000b    ;color 0
          db 10000000b, 01000000b, 00100000b, 00010000b    ;color 1
          db 00001000b, 00000100b, 00000010b, 00000001b    ;color 1

dseg      ends

; ----------------------------------------------------------------------
;                          Main Program Logic
; ----------------------------------------------------------------------
comgroup group    cseg, dseg

cseg      segment   para public 'CODE'
          assume    cs:comgroup, ds:comgroup, es:comgroup

;***********************************************************************
; PLOTDOTH
;
; Hi-resolution graphics pixel-plot function.
; Uses look-up tables for fastest operation.
;
;  plotdot(x,y,color)
;    int x;         horizontal (0-639) not value-checked
;    int y;         vertical (0-199) not value-checked
;    int color;     color for dot (0 to 1)
;
;    Assumes that ES is pointing to the segment address B800h, the CGA
; video memory.
;
;***********************************************************************
plotdoth proc   near              ;SMALL MODEL ONLY
         pop    si                ;Fetch return addr--this technique is faster
         pop    bx                ;Get X value-faster than stack-relative access
         pop    di                ;Get Y value
         pop    cx                ;Get color
;
; *** BEGIN Address Calculation ***
;
         shl    di,1              ;Index into row address lookup table
         mov    di,row_tbl[di]    ;DI points to start of row
         mov    ax,bx             ;Copy the X value
         mov    cl,3
         shr    ax,cl             ;Divide by 8 to get the byte offset
         add    di,ax             ;DI points to the byte to modify
;
; *** END Address Calculation ***
;

         mov    al,es:[di]        ;Get current screen byte

         and    bx,7              ;Get pixel-offset in byte (0 thru 7)
         and    al,mask_tbl[bx]   ;Mask a "hole" into the current byte
;
;-------- Index into the color table.
;
         and    cx,1              ;Make sure it's a valid color
         shl    cx,1
         shl    cx,1
         shl    cx,1
         add    bx,cx             ;Index is (COLOR * 8) + (X mod 8)
         or     al,color_tbl[bx]  ;Fill the "hole" with the selected color

         mov    es:[di],al        ;place modified byte back into screen
;
; Now for a really quick way of returning to the caller, as
; artificial NEAR return. No need for a RET.
;
         jmp    si
plotdoth endp
cseg     ends
         end
```

You can deliver it before 199~~1~~2.

New Version 2.2
Compatible with DOS 4.0
Uses less memory, runs faster, and adds communications support

It's coming. An era of more powerful PCs. Easier to use PCs. With graphics and character-based programs working side by side. Talking to each other. Multitasking. Windowing. Menuing. Mousing. Letting your customers get their work done easier and faster.

Sell it all now.

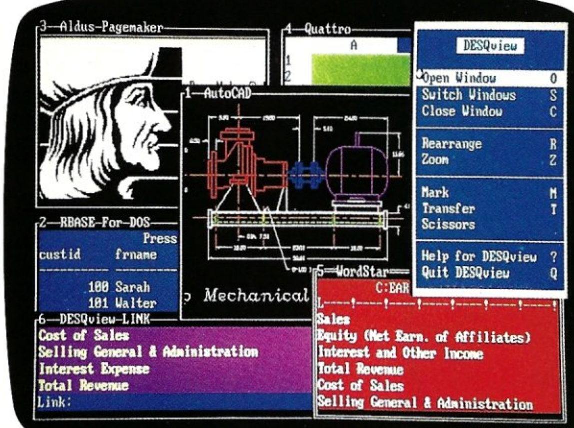
DESQview™ is the operating environment that gives DOS the capabilities of OS/2.™ And it lets your customers, with their trusty 8088, 8086, 80286, or 80386 PCs, leap to the productivity of the next generation. For not much money. And without throwing out their favorite software.

Add DESQview to your customers' PCs and it quickly finds their programs and lists them on menus. So they can just point to the program, using keyboard or mouse, to start it up. DESQview knows where that program lives. And what command loads it.

For those who have trouble remembering DOS commands, it adds menus to DOS. It even lets them sort files and mark specific files to be copied, backed-up, or deleted— all without having to leave the program they're in.

Best of all, DESQview accomplishes all this with a substantial speed advantage over any alternative environment.

For programmers, DESQview's API, with its strengths in inter-task communications and multi-tasking, brings a quick and easy way to adapt to the future. With the API's mailboxes and shared programs, programmers are able to design programs running on DOS with capabilities like those of OS/2.



One picture is worth a thousand promises.

Multitasking beyond 640K.

When your customers want to use several programs together, they don't have to leave their current program. Just open the next program. View programs in windows or full screen. Open more programs than they have memory for. And multitask them. In 640K. Or if they own a special EMS 4.0 or EEMS memory board, or a 386 PC, DESQview lets them break through the DOS 640K barrier and multitask. For instance, they can start 1-2-3 calculating and tell Paradox to print mailing labels while they're writing a report in Word Perfect, or laying out a newsletter in Ventura Publisher, or designing a building in AutoCAD.

DESQview even allows them to transfer text, numbers, and fields of information between programs.

Fulfill the 386 promise.

For 80836 PC users, DESQview becomes a 386 control program when used in conjunction with Quarterdeck's Expanded Memory Manager (QEMM)-386—giving faster multitasking as well as virtual windowing support.

And when you use DESQview on an IBM PS/2™ Model 50 or 60 with QEMM-50/60 and the IBM Memory Expansion Option, DESQview gives multitasking beyond 640K.

Experts are voting for DESQview. And over a million users, too.

If all of this sounds like promises you've been hearing for future systems, then you can understand why so many VARs and system integrators have chosen DESQview.

PC MAGAZINE EDITOR'S CHOICE NOV. 24, 1987

And why *PC Magazine* gave DESQview its Editor's Choice Award for "The Best Alternative to OS/2," why readers of *InfoWorld* twice voted DESQview "Product of the Year" why, by popular vote at Comdex Fall for two years in a row, DESQview was voted "Best PC Environment" in *PC Tech Journal's* Systems Builder Contest.

INFO WORLD 1986 AND 1987 PRODUCT OF THE YEAR

DESQview lets you sell it all now.

DESQVIEW SYSTEM REQUIREMENTS:

IBM Personal Computer and 100% compatibles (with 8086, 8088, 80286, or 80386 processors) with monochrome or color display; IBM Personal System/2• Memory: 640K recommended; for DESQview itself 0-145K• Expanded Memory (Optional): expanded memory boards compatible with the Intel AboveBoard; enhanced expanded memory boards compatible with the AST RAMpage; EMS 4.0 expanded memory boards•Disk: two diskette drives or one diskette drive and a hard disk•Graphics Card (Optional): Hercules, IBM Color/Graphics (CGA). IBM Enhanced Graphics (EGA), IBM Personal System/2 Advanced Graphics (VGA)• Mouse (Optional): Mouse Systems, Microsoft and compatibles• Modem for Auto-Dialer (Optional): Hayes or compatible• Operating System: PC-DOS 2.0-4.0; MS-DOS 2.0-4.0• Software: Most PC-DOS and MS-DOS application programs; programs specific to Microsoft Windows 1.03-2.1, GEM 1.1-3.0, IBM TopView 1.1• Media: DESQview 2.0 is available on either 5-1/4" or 3-1/2" floppy diskette.

Quarterdeck
150 Pico Blvd., Santa Monica, CA 90405
(213) 392-9851 Fax: (213) 399-3802

Trademarks are property of their respective holders: IBM, OS/2, PS/2, 1-2-3, Paradox, Word Perfect, Ventura Publisher, AutoCAD, Intel, Above Board, AST, RAMpage, Advantage, Hercules, Mouse Systems, Hayes, Microsoft, Windows, TopView.

CIRCLE 101 ON READER SERVICE CARD.

of a look-up table. Listing 1 is dedicated to plotting dots on the CGA adapter in high-resolution graphics mode, but it is easily modified for the EGA card.

On the CGA card, even-row pixel values reside within the first 8 kilobytes (2000 hex) of memory in 80-byte (50 hex) increments, while the odd-row pixel values reside within the second 8 kilobytes. For example, pixels in row 0 occupy locations 0 – 4F. Pixels in row 1 occupy locations 2000hex – 204Fhex. The macro assembler REPT pseudo-op conveniently builds the ROW_TBL according to this hardware addressing scheme. The row address calculation for the EGA card can be calculated by changing the DW ADDR,ADDR+2000h to DW ADDR,ADDR. The code of interest in between the BEGIN-END Address Calculation comments in Listing 1. By similar application of the look up table technique, pixel mask generation was accomplished.

The timings I performed on Professor Dwyer's routine and mine indicate that my routine executes in 71 percent of the time his program did. The inclusion of the table-driven pixel mask generation will net another 3 percent to bring the execution time down to 68 percent. I invite others to improve even more upon the work of Professor Dwyer and myself.

Robert Zigon
Indianapolis, Indiana

Author's note: Mr. Zigon's suggestion for speeding up the timing of the EGA-DOT procedure is well taken. Although his code was for the CGA adapter, I am sure that the table look-up concept could be applied to the code for the EGA adapter as well. Unfortunately applying it to the machine-independent library from which the routines in our article were taken may be another story. As to the use of VRAM, yes, I am familiar with its current usage, especially since I am on the Texas Instruments Graphics Products mailing list. In my classes, I use MPRAM to mean "multi-ported RAM."

Take the Shortcut Home

Dear Micro/Systems:

I enjoyed Brian Brown's letter and solution to users getting lost via directory wandering ("Lost in Novell Gnet? You Can Go Home Again," *Micro/Systems*, July 1988, page 12). I would like to suggest an alternative approach. For this alternate solution to work, you must remove the following line:

```
SET HOME="D:\\UO\\BRIANB"
```

from the login script in Listing 2 and replace it with the following two lines:

```
SET HOMEDRV=D:
SET HOMEDIR="\\UO\\BRIANB"
```

The following batch file, GETBACK.BAT, can now be used:

```
%HOMEDRV%
CD %HOMEDIR%
```

Augustine J. Cannata
C-Tree Systems, Inc.
Stanhope, New Jersey

Getting the Lay of the LAN

To the reader,

As the cover banner explains, *Micro/Systems Journal* is devoting its editorial coverage exclusively to "LAN and Multiuser PC Integration." We want to provide the kind of practical, applications-oriented information that will help you improve your own LAN technology.

But to keep our articles targeted to your needs, we need feedback. Please write to us with comments or suggestions on the articles you have read or would like to read in *Micro/Systems Journal*. Rating articles through the reader service card also tells us what you think is hot and what's not. If you tell us what interests you, we can seek out the experts and get them to write the kinds of articles that will help you solve your own PC connectivity problems.

The Editors □

by Don Libes

Achieving Portability

Portability. It is one of the best things about C. It is also one of the most difficult things to achieve in a C program. It should be obvious, but I'll say it anyway; Programs are not portable just because they are written in C. The portability of C programs is one of those myths, like the portability of UNIX programs.

This column discusses portability, gives some guidelines, presents some examples, makes some suggestions, and recommends some books on the subject.

Much of what constitutes "portable programming practice" is common to the tenets of "good programming practice." For example, you should always check return codes. Unfortunately, code can be very difficult to read when conditions surround every statement. If the code is going to appear as a stand-alone example in a piece of text, and the error codes are not relevant, that *must be noted.* Only then is it acceptable for the error checking to be omitted. However, the production code better have the error checking.

In the same fashion, nonportable code may be presented, but *only if it is so noted.* I have often presented code that works on a subset of machines but always identified it as such. If you do similarly in production code, make sure your comments explain any potential problems. To do otherwise would be cruel.

You should always make an effort to produce portable code. Only if it means you would sacrifice efficiency or functionality should you write nonportable code. And then, you should try to provide a hook so the programmer who has to get your garbage running on a different machine doesn't have to rewrite the thing.

Your code *will* be ported to other machines. Unless you live on a desert island (in which case you won't be getting this magazine), it is inevitable. So, face the music. From another viewpoint, consider that you *will* use code from another environment eventually. To admit otherwise means that you will be stuck using the same machine you are currently using for the rest of your life. In the computer field, this is not a reality.

Why Is Portable Code So Important?

One of the things that C has become famous for is making it possible to write portable system code. For example, 99 percent of all UNIX system software is written in C. This is one of the reasons it is so much easier to port UNIX programs than, say, CP/M programs, 99 percent of which were written in 8080 and Z80 assembler. Many UNIX programs, for example, run under DOS with almost no change.

DOS programs imply the 80x86 architecture, and one might think that its corresponding assembler language would be portable. To a certain degree this is true, however, few people write 80x86 assembler. While assembler programs can be faster than C programs, the speed improvements simply aren't worth all the drawbacks. If you don't need the speed, there is almost no reason why you can't code in C. By writing in C, you increase the likelihood that your programs can be easily moved to other operating systems or machine architectures.

Portability is not always easy to achieve and there are differing degrees to which various people believe in portability. Many aren't concerned in the least. Should you be? Is the price too high to pay? The answer is a qualified maybe. The reason is because portable code has disadvantages as well as advantages.

The primary advantage is that code is easier to port (this is by definition). Portable code usually has the additional benefit that it is better written code. Almost always, more thought will have gone in to low-level code writing (as opposed to design) in order to achieve portability.

The drawbacks are obvious as well. Producing portable code *the first time* is more expensive, especially if you are not going to port it. Another drawback is that code can often be unreadable or unwieldy, although more often it is nonportable code that is unreadable and unwieldy. Lastly, portable code often sacrifices environment-dependent hacks that are more efficient than slower-but-portable techniques.

There are other issues, but elaborating on them hardly sways the conclusion. The fact is, if there is any likelihood the code will be ported, it is worth the immediate expense to make it portable in the first place. This covers the majority of C code.

What's Left?

What am I willing to code nonportably? Parts of the guts of a C compiler are inherently nonportable, such as function entry and exit, and certain run-time functions like *setjmp( )*. System boot code, drivers that cannot afford the overhead of C function calls such as the clock, and other time-critical code are all examples of software that usually has to be written in assembler.

Lastly, if you are optimizing code, you may find one or two *hot spots* that allow large speedups with just one or two lines of assembler. However, cavalier abuse of this can be a drawback rather than an advantage. In the August 1988 issue of *Micro/Systems*, I discussed how *strcpy( )* was sped up by rewriting it in assembler for the VAX. While this worked for the first VAX model, the resulting code was actually slower than the original C code on some later VAX models.

Of course, before giving up on C, try rethinking the code. If an algorithm can be improved, this will buy more speed than rewriting it in assembler. For example, you can optimize your bubble sort until the end of time, but even the most simplistic C implementation of quick sort will be able to beat the assembler-coded bubble sort.

Don Libes is a computer scientist based in the Washington, D.C., area. He is also the author of Life With UNIX *published by Prentice-Hall.*

What Does Portability Mean?

There are portability issues at many different levels. Here are four broad groupings:

1) *Machine and language data and instruction formats.* Byte ordering is an excellent example of this. Byte ordering is completely inconsistent from one architecture to another. Programs should never depend on byte ordering. (See *Micro/Systems Journal*, July/August 1987, for an entire column devoted to byte ordering.)

There are many other problems that crop up at this level. For example, your programs should not depend on code that assumes a particular character set, or code that assumes one of signed or unsigned *chars*.

Note that *lint* will catch most of these problems. There is no excuse for not using *lint*.

2) *Low-level operations.* Null pointer dereferencing is a good example of nonportability. On a VAX, a null pointer can be dereferenced (and the result is always 0). This seems strange, but a lot of sloppy code has actually been written that depends upon this.

Most machines do not allow users to dereference location 0. On unmapped systems, it is usually reserved for interrupt vectors anyway, and on mapped systems, it simply isn't available to the process. Such references produce address faults.

Other examples of nonportable, low-level operations is accessing memory in ways that violate alignment constraints. For example, the 68000 does not allow word access beginning on odd-byte boundaries.

3) *C language syntax and semantics.* As you are probably aware, there is an effort to produce an ANSI C standard. Prior to this, compiler implementors never had a final authority. Some followed Kerningham and Ritchie, some followed *pcc*, and others followed their own imagination. Because of this, there are many discrepancies among C compilers.

For example, some C compilers require internal names to be unique within seven characters. Variables with longer names can be used, but only the first seven are actually entered into the symbol table. The solution is to keep names unique within the first seven characters, even if you use longer names.

Another example is to avoid passing the negative integers to the modulus operator (%). There are two common but different results in this case, and the problem is easy to avoid.

4) *Operating system environment.* Differing environments can cause many problems to C programs. For example, a common restriction is that externally visible names must be unique within six characters, regardless of case. This problem is imposed by many linkers whose specifications were set in stone years ago.

Another problem is system calls that are not portable from one operating system to another. A generally useful guideline is to avoid all system calls and use library functions. Since these are typically higher-level and easier to use, this should cause little problem.

Of course, libraries corresponding to very specialized tasks, such as window systems and networks, can make it exceedingly difficult even to find a common set of library calls.

Some General Guidelines

Now that we have listed the issues affecting portability, let's look at some rules to make coding for portability easier.

1) *lint* is particularly helpful with low-level coding. It specifically looks for portability problems that can be detected at the syntactic level and is very comprehensive.

2) Reduce the use of magic numbers as much as possible. Generally, the only numbers that should appear in your code are 0, 1 and -1. The rest should be in *#defines*.

3) Avoid going behind the C compiler's back. This means not assuming knowledge of structure layout or byte ordering, and avoiding excessive casting when you should be using a union.

4) Avoid questionable practices, even if they aren't behind the compiler's back. Never assume that *char* is signed. Never assume you can dereference a null pointer. Be very careful when declaring variables to be *int*. *int* is a convenience for the architecture and provides few guarantees about how big it will be.

5) Be sure that function arguments match types. For example, if a routine expects a *long*, make sure it gets one. Never pass untyped nulls that are going to be used as pointers without casting.

6) Try porting your software to several different architectures. This is an amazingly useful and sometimes painful way to learn about the real world of computing.

7) Isolate nonportable code in separate files, or through pre processor directives such as *#if* and *#ifdef*.

Books on Portability

This column discusses only the tip of the iceberg in the area of portability. Porting expertise can only be gained through close reading of many manuals, years of experience trying to actually move software from one environment to another, and occasional visits to the psychiatrist after finding out that your favorite coding practice is nonportable and you've got to change hundreds of programs.

I can recommend several books that you might like to read for further information on writing portable code. One that I have mentioned several times in this column is *C: A Reference Manual* by Samuel Harbison & Guy Steele (Prentice-Hall). This book is not really a text on portability, but because it is so comprehensive in scope, the authors continually mention how many compilers diverge from each other. Many examples of how to code portably are presented.

Portable C and UNIX System Programming by J. E. Lapin (Prentice-Hall) is not as comprehensive about the C language as Harbison and Steele, but is oriented more towards information about libraries and environments. The book covers the differences between calls in System V, Berkeley, Xenix, and several other UNIX variants.

Another book on portability that will be available by the time you read this column is *Portability and the C Language* by Rex Jaeschke (Hayden). I have not seen the book, but I have read many excellent articles by Rex, and I have high hopes for the book.

Conclusion

Portability is not guaranteed just by writing in C. You must spend a large amount of effort to create portable code, but the results are almost always worth it. Portable code can be moved to other environments with no cost, while nonportable code can require complete rewriting, and substantial redesign at worst.

Keep in mind some of the guidelines I have suggested as you write your next program. There is no reason to write nonportable code if you don't have to. The next programmer to receive your code will thank you. And don't forget to use *lint*!

Many thanks to Martin Minow, Guy Harris, Joe Yao, Dan Kegel, and Henry Spencer for suggesting some of the portability guidelines discussed in this column. □

Did you find this article particularly useful? Circle number 1 on the reader service card.

by Stephen R. Davis

Doubly Linked Lists

Many computer educators and the books they write spend a great deal of time instructing students in sophisticated data storage techniques to improve computer system performance. Few of these educators spend much time instructing students in techniques to get these data structures up and running in a reasonable amount of time. In this column we will look at the simple, but very effective, signature field technique for detecting errors in linked lists. Fortunately this technique is directly applicable to other data structures that use pointers.

Before we begin the discussion, let's take a quick look at an interesting Pascal product, Pecan's UCSD Pascal. Unlike other products examined in this column, this one does not deal directly with Turbo Pascal. Instead, UCSD provides an alternative to Turbo Pascal with some interesting differences.

Today's Product: UCSD Pascal

UCSD Pascal, as its name implies, began its life at the University of California at San Diego in the early years of CP/M, long before MS-DOS and the PC. At that time, no one machine or processor had gained anything like dominance of the market. In such an environment, it was difficult to build a language compiler since the expected audience could only be a small percentage of the total market.

The developers of UCSD Pascal had an idea, however. Since no one processor was dominant, they would invent yet another processor. Although it never physically existed, this virtual processor had an instruction set called p-code that was optimized for their compiler. To implement UCSD Pascal on a machine, you had to write a p-code interpreter for the target machine to turn it into the virtual p-code machine required by the compiler. Since p-code was kept simple, these p-code interpreters were relatively easy to implement. UCSD Pascal quickly became available in essentially unchanged form on a wide variety of machines.

A compiler is useless without support software, so the UCSD system evolved, complete with all the normal development tools, including a command interface. All these were implemented in p-code so they all ran under the same p-code interpreter developed for Pascal, providing essentially the same environment on all machines.

Since that time, Pecan Software Systems has bought and now markets the UCSD system. The current version of UCSD includes compilers for a variety of languages such as Modula-2, FORTRAN-77, BASIC, and C. Essentially the same UCSD environment, programming tools, and compilers are supported on the IBM PC, Atari ST, Amiga, VAX, Macintosh, Apple II, and CP/M. In addition, multiuser versions of the UCSD that will support up to four users are available.

UCSD Pascal includes the standard extensions that serious programmers require, including units. A plus that I personally find exciting is the inclusion of intrinsic support for multitasking. Declaring a procedure to be of type *process* makes it available to be instantiated as independent processes (as many as desired) via the *Start* system call. Processes may be attached to interrupts or *Signal*ed and semaphored in a fashion reminiscent of UNIX.

With all this going for it, why does the UCSD System not enjoy greater popularity? The problem is primarily one of performance. Although executing UCSD-generated code through a p-code emulator provides great flexibility, it also extracts a penalty in performance. To address this to some degree, Pecan provides a native code generator that accepts the UCSD p-code generated by their compilers as input and outputs the assembly language of the native machine. This native code runs much faster than the p-code. However, because the original compiler was not built around the native assembly language, optimization may be a problem. Still, the UCSD system could well be highly desirable to programmers who intend to execute their programs in several different environments.

Today's Topic: Signature Fields

As mentioned, many programmers spend insufficient attention to the problem of actually implementing complex data structures, especially those involving linked lists. A program that has corrupted a linked list pointer is much like a train that has left its tracks. The program will eventually crash (the chance that it will accidentally fall back onto the tracks is negligible), but it may crash miles from the point where the pointer was actually lost. Early detection of the errant pointers greatly decreases the difficulty involved in implementing linked list structures.

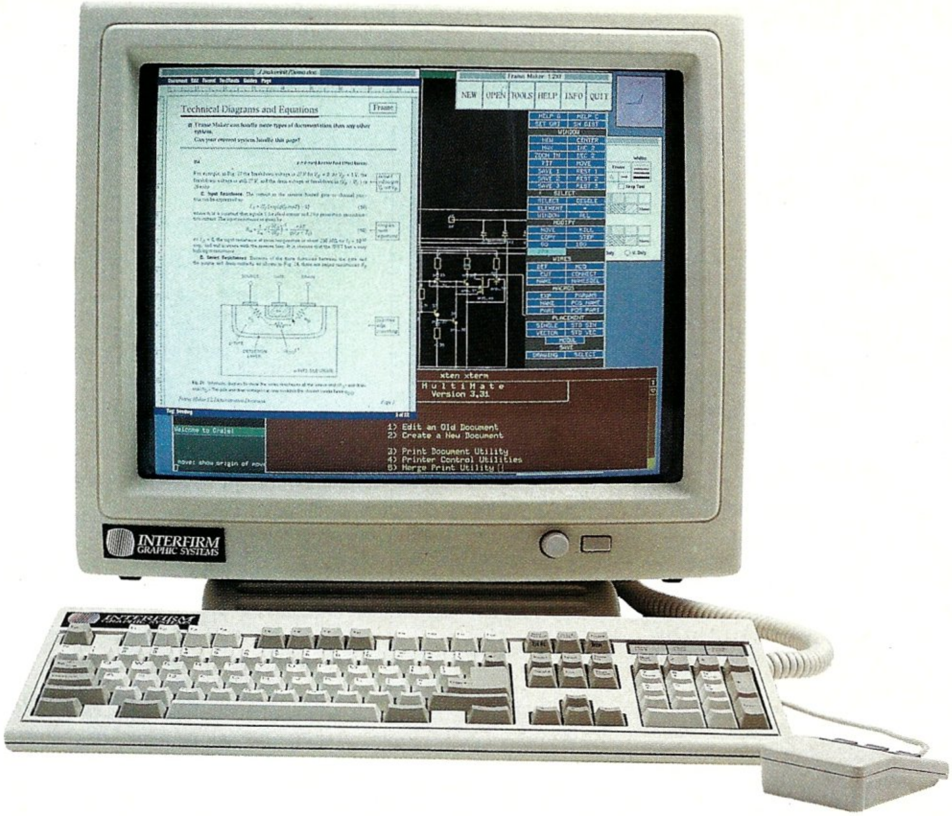
One way to detect pointer problems is with signature fields. In this method, a field in the record is set aside to hold a unique identifier of the record type. Every time a pointer to this record type is passed to a function, the function can check this field against the expected value. If the fields do not match, the function knows that the pointer does not point to what the user thinks it points and can stop immediately. This isolates the problem very close to the derailment, thus simplifying the programmer's debugging problem.

Listing 1 shows a Turbo 4.0 unit that implements a doubly linked circular list. This list uses an empty record, called the head record, to hold the address of the first and last members of the list. Each member contains the usual forward and backward pointers defined in the record type *ListElem*; however, added to these is a 16-bit signature field called *Sigfield.*

In use, the user initializes a new linked list by calling the function *InitList*. This routine allocates a record for the head record from an array of such records. It then generates a signature field by concatenating a nibble containing the index of

Stephen Randy Davis is a technical editor for Micro/Systems Journal *and a programmer for a defense contractor in Greenville, Texas.*

Get a window on the future . . .



without giving up the present

At last, you can have the benefits of UNIX™ V.3 and high resolution graphics through X-Windows—without sacrificing the investment you've made in your DOS based 80386™ system.

The Easy Way to X-Windows

With the XSUBSYSTEM™ from Interfirm Graphic Systems, you'll have everything you need to convert your 80386-based machine into a powerful UNIX-based graphics workstation. Suddenly, you'll have all of the advantages of a multi-user, multi-tasking system—plus the incredible convenience of DOS programs runnng side-by-side with UNIX.

The XSUBSYSTEM Includes:

- 19″ High resolution color or monochrome display
- Graphics controller
- UNIX System V.3 from Interactive Systems Corp.
- The X Window System software
- Optional hard disk with all software pre-installed
- Optional TCP/IP Ethernet controller and software

INTERFIRM
GRAPHIC SYSTEMS
3940 Freedom Circle
Santa Clara, CA 95054
(408) 970-7070
800-338-4513

The Future of 386 Based Graphics

The XSUBSYSTEM gives you the best of both worlds - you get to keep both your hardware and your DOS software while you upgrade to a UNIX V.3/X-Windows System.

For example, with Interfirm's XSUBSYSTEM, you can take data created in a DOS application and paste it into a document that you are creating with the Frame Maker™ desktop publishing system.

It's the most sensible way we know to give your 386 based system the power of a true graphics workstation - at a sliver of the price.

At Interfirm Graphic Systems, we are pioneering UNIX-based high-performance graphics, utilizing the X Window System and other emerging non-proprietary standards. When you look to the future of desktop computing, look to Interfirm.

Trademarks: UNIX-Bell Laboratories; XSUBSYSTEM - Interfirm Graphic Systems; MS DOS - Microsoft; X Window System - M.I.T.; 80386-Intel Corp; Frame Maker-Frame Technology Corp.

CIRCLE 109 ON READER SERVICE CARD

Listing 1. Unit LList

```
Unit LList;
{LList -
    Impliment doubly linked circular list with signature checking
    of fields. List is initialized by calling InitList, which
    returns 16-bit signature field uniquely identifing list. This
    signature field must be presented to other routines for
    verification. Individual elements are initialized by passing
    them to InitElem before adding them to a list.
}
Interface
    Type
        Signature = integer;
        ElemPtr  = ^ListElem;
        ListElem = record
                      sigfield  : signature;
                      prev, next: elemptr;
                   end;
    Const
        Numlists = $10;

    {Link -
        add 'element' to the list immediately after 'before'. If
        'before' is a nil, add 'element' to the end of the list.
        Return true if successful, false if not.}
    Function Link (element, before : pointer; sig : signature) : boolean,
    {Unlink -
        remove 'element' from the list. Return true if successful.}
    Function UnLink (element : pointer; sig : signature) : boolean;

    {Next, Previous -
        Given an entry 'element', return the address of the next
        or the previous entry in the list. Given a NIL, returns
        the first element in the list and returns a NIL when the
        list has been exhausted}
    Function Next     (element : pointer; sig : signature) : pointer;
    Function Previous (element : pointer; sig : signature) : pointer;

    {Head -
        Returns the head pointer of a list. Returns nil on error.}
    Function Head (sig : signature) : pointer;

    {InitElem -
        Initialize a new record 'element' to be added to a list.}
    Procedure InitElem (element : pointer; sig : signature);

    {InitList -
        Initialize a new list for use. Returns a signature field
        which must be used in every subsequent access of that list.
        Returns a 0 in the event of an error.}
    Function InitList : signature;

Implementation
    Var
        headspace : array [1..NumLists] of listelem;
        index : integer;

    Function Link (element, before : pointer;
                            sig : signature) : boolean;
    Var
        ok : boolean;
        after : elemptr;
    Begin
        ok := true;

        {check 'element' to make sure that it is valid
         entries to be added to the list}
        with elemptr(element)^ do
        begin
            if sigfield <> sig then ok := false;
            if prev     <> nil then ok := false;
            if next     <> nil then ok := false
        end;

        {now check 'before'. If it is nil, tack entry onto end}
        if before = nil then
            before := elemptr(Head(sig))^.prev;
        if elemptr(before)^.sigfield <> sig then ok := false;

        {if no problems, then go ahead and add them}
        if ok then
        begin
            elemptr(element)^.prev := before;
            after := elemptr(before)^.next;
            elemptr(before )^.next := element;

            elemptr(element)^.next := after;
            after^.prev := element
        end;
        Link := ok
    End;

    Function Unlink (element : pointer; sig : signature) : boolean;
    Var
        before, after : elemptr;
Begin
    if elemptr(element)^.sigfield <> sig then
        UnLink := false
    else
    begin
        before := elemptr(element)^.prev;
        after  := elemptr(element)^.next;
        before^.next := after;
        after ^.prev := before;
        elemptr(element)^.prev := nil;
        elemptr(element)^.next := nil;
        UnLink := true
    end
end;

Function Next (element : pointer; sig : signature) : pointer;
Begin
    {Given a nil, start with the head element}
    if element = nil then
        element := Head (sig);

    {Now get the next element}
    if elemptr(element)^.sigfield <> sig then
    begin
        {Declare an error here!}
        write ('Pointer error!');
        element := nil
    end
    else
        element := elemptr(element)^.next;

    {If we ran off the end, return a nil}
    if element = Head (sig) then
        element := nil;
    Next := element
End;

Function Previous (element : pointer; sig : signature) : pointer;
Begin
    if element = nil then
        element := Head (sig);
    if elemptr(element)^.sigfield <> sig then
    begin
        {Declare an error here!}
        write ('Pointer error!');
        element := nil
    end
    else
        element := elemptr(element)^.prev;
    if element = Head (sig) then
        element := nil;
    Previous := element
End;

Function Head (sig : signature) : pointer;
Var
    hdptr : elemptr;
Begin
    hdptr := @headspace[sig and $000f];
    if hdptr^.sigfield <> sig then
        hdptr := nil;
    Head := hdptr
End;

Procedure InitElem (element : pointer; sig : signature);
Begin
    with elemptr(element)^ do
    begin
        sigfield := sig;
        prev     := nil;
        next     := nil
    end
end;
   Function InitList : signature;
   Var
       P : elemptr;
       S : signature;
   Begin
       if index > numlists then
           S := 0
       else
       begin
           {Init the header record}
           P := @headspace[index];
           P^.next := P;
           P^.prev := P;

           {Make the Least Significant Nibble the index and
            the rest random}
           S := (Random ($4000) and $fff0) + index;
           P^.sigfield := S;
           index := index + 1
       end;
       InitList := S
   End;

Begin
   index := 1;
   Randomize
```

the head record to three nibbles generated randomly. *InitList* returns this signature value to the user code. This number must be safeguarded since it will be required for all future accesses of this list.

Elements to be added to the list can be allocated off the heap via the New Pascal call or from a previously defined array. Either way, these records must be initialized via the routine *InitElem*. Initialized elements are then added to a list with the *Link* routine and removed from the list with the *UnLink* routine while sequencing through the list is handled with the *Next* and *Previous* functions.

Listing 2 shows a simple test routine designed to demonstrate the use of some of the unit's functions. While it is not comprehensive, it does show the proper use of *InitList, InitElem, Link,* and *Next.* Single-stepping this program with a source code debugger such as *TDebugPlus* will help reveal the inner workings of the linked list unit.

In use, this unit is much more resilient than linked list units that do not use signature fields. If the user code presents the unit with a pointer that is errant or that does match the signature field presented, this unit will indicate the problem immediately. Try it. Make this trick a permanent part of your grab bag of programming techniques. □

. . . unit is much more resilient than linked lists that do not use signature fields.

Listing 2. Test of LList Module

```
{ Simplistic test of LList module }
Program Test;
uses llist;

Type
     dataptr = ^data;
     data = record
            header: listelem;
              {data record must have a header}
            block : array [1..100] of integer
          end;
Var
     dsig    : signature;
     current : dataptr;
     i,j     : integer;

Procedure PrintOut;
Var
     listptr : dataptr;
Begin
     listptr := nil;
     while Next (listptr, dsig) <> nil do
     begin
          listptr := Next (listptr, dsig);
          Writeln (listptr^.block[1])
     end
End;

Begin
     dsig := InitList;
     for i:= 1 to 10 do
     begin
          New (current);
          InitElem (current, dsig);
          current^.block[1] := i;
          if not Link (current, nil, dsig) then
               writeln ('Link error on index ', i)
     end;
     PrintOut
End.
```

Did you find this article particularly useful? Circle number 2 on the reader service card.

Product Information

Pecan Software Systems
1410 39th Street
Brooklyn, NY 11218
(800) 637-3226
Circle reader service #250

Troubleshooting Local Area Networks Using the OSI Model

by Mark A. Miller

This article reviews the OSI reference model for computer topology, outlines the OSI model in a LAN context, and then describes how you can use the model to facilitate finding network faults.

The Open Systems Interconnection (OSI) reference model developed by the International Standards Organization (ISO) is a layered structure that has been widely used to provide a theoretical benchmark for comparing computer networks. In this article, I will first review the functions of each layer, and then address the way in which the model applies to a local area network (LAN). Second, I will investigate how our theoretical knowledge of the LAN model can be put to practical use by guiding our selection of appropriate tools to troubleshoot a LAN that is either not working or operating in a less than optimal manner.

Mark A. Miller is president of DigiNet Corporation, a data communications engineering firm based in Denver.

The OSI Model Reviewed

Figure 1 shows the OSI reference model as a seven-tiered cake. Information from higher layers becomes encapsulated in the lower layers prior to transmission across the cable or transmission medium. The functions of each individual layer are:

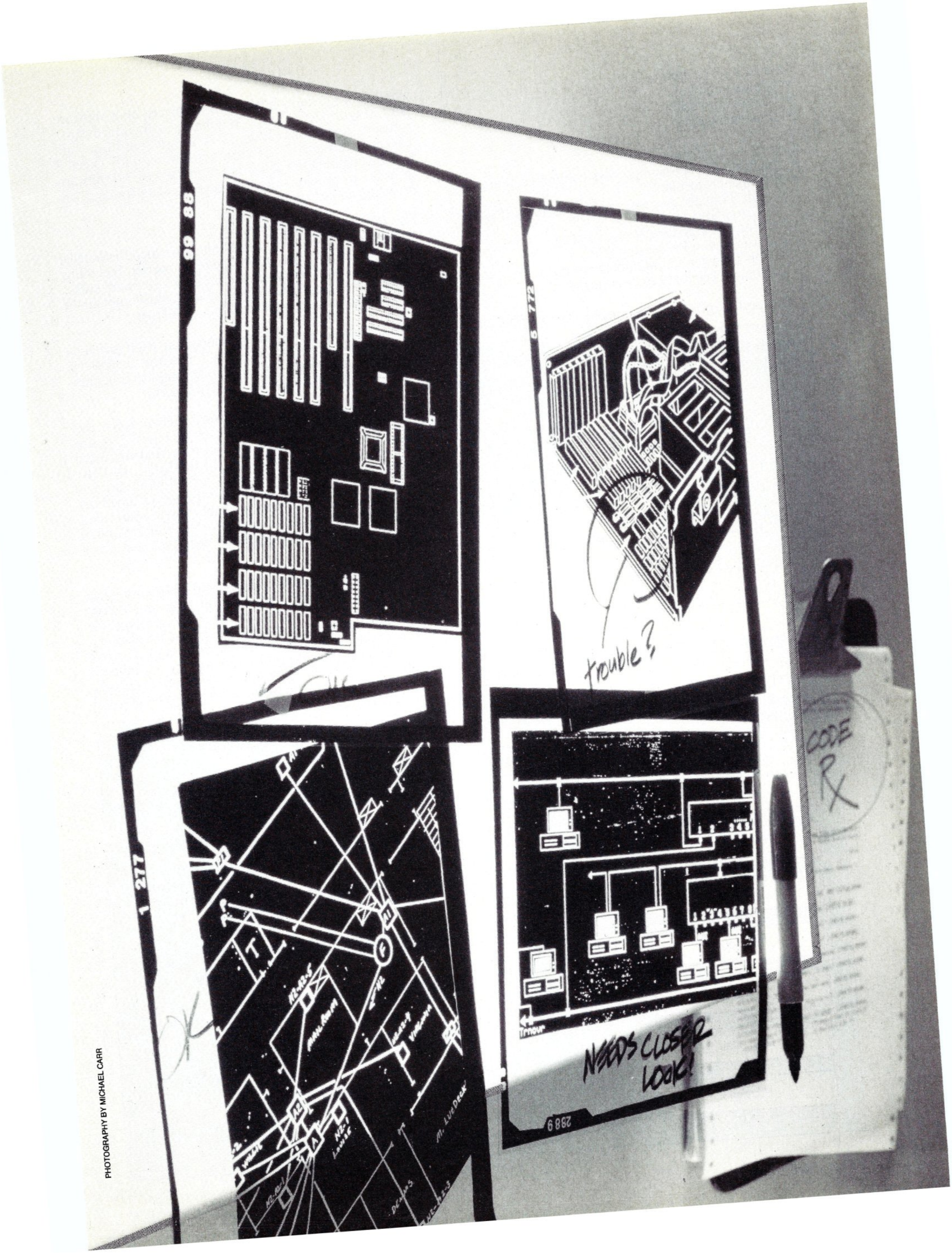
Layer 1 — Physical
- Voltages and electrical pulses
- Media and the media interface
- Line discipline (half- or full-duplex)
- Pin assignments

Layer 2 — Data Link
- Framing
- Error control
- Data transparency

Layer 3 — Network
- Routing within the communications subnet
- Sequenced delivery
- Congestion control

Layer 4 — Transport
- Host-to-host connection establishment
- Message segmentation and multiplexing
- Reliable end-to-end delivery
- End-to-end flow control

Layer 5 — Session
- Connection establishment and termination
- Dialogue discipline (who speaks when, how long, and so on)
- Synchronization between end-user tasks



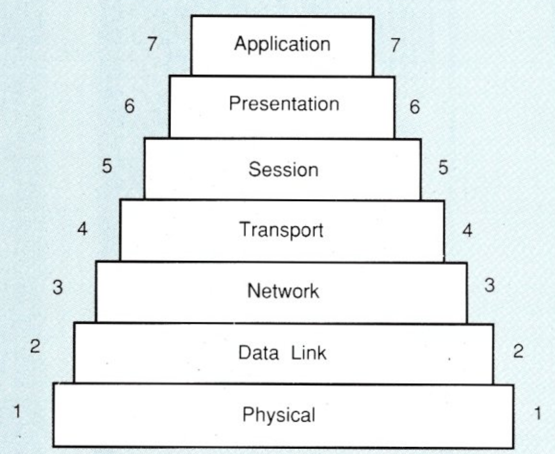
PHOTOGRAPHY BY MICHAEL CARR

Layer 6 — Presentation
- Syntax definition
- Text compression
- Data encryption

Layer 7 — Application
- Application process-to-process communication
- Example protocols:
  - File Transfer, Access, and Management (FTAM)
  - Virtual Terminal (VT)
  - Message Handling System (MHS)
  - Transaction Processing (TP)
  - Job Transfer and Manipulation (JTM)
  - Remote Database Access (RDA)

The first step in LAN troubleshooting is determining whether a network failure is a hardware or a software fault (Figure 2). Figure 1 delineates the hardware/software definitions. Layer 1 is clearly hardware and Layers 3 through 7 are clearly software, with Layer 2 being a combination of hardware, firmware, and software in order to provide such functions as data framing, addressing, and error control.

Figure 1. The Open Systems Interconnection (OSI) reference model



The OSI Model and LANs

As mentioned previously, the OSI reference model is equally applicable to all types of computer networks, including mainframe and minicomputer architectures and LANs. Figure 3 shows how the model applies to the components of a generic LAN; the relationship is as follows:

Layer 1: Includes the transmission medium itself (twisted-pair, coaxial, or fiber-optic cable) plus the necessary connectors, amplifiers, and so on to accomplish transmission and reception of the raw bits (1s and 0s).

Layer 2: Implemented on the Network Adapter Card, the principal responsibility within a LAN at Layer 2 is the framing of the higher-layer data into a well-defined (or predictable) sequence. Ethernet, Token Ring, StarLAN, ARCNET, and other frame formats are all different, and each gives the LAN its principle identity. Other functions at Layer 2 include source and destination node addressing to identify the sender and intended receiver of the frame, plus a cyclical redundancy check (or CRC) for error control.

Layers 3 – 5: Implemented in NetBIOS (the Network Basic Input/Output System) developed by IBM and Sytek. NetBIOS provides the session (Layer 5) functions for establishing logical connections between LAN workstations; it omits some transport (Layer 4) and network (Layer 3) functions that are not required by a local area network but that would be found in a wide area network, such as node routing, end-to-end acknowledgments, and

Figure 2. Implementing the OSI Model in hardware and software

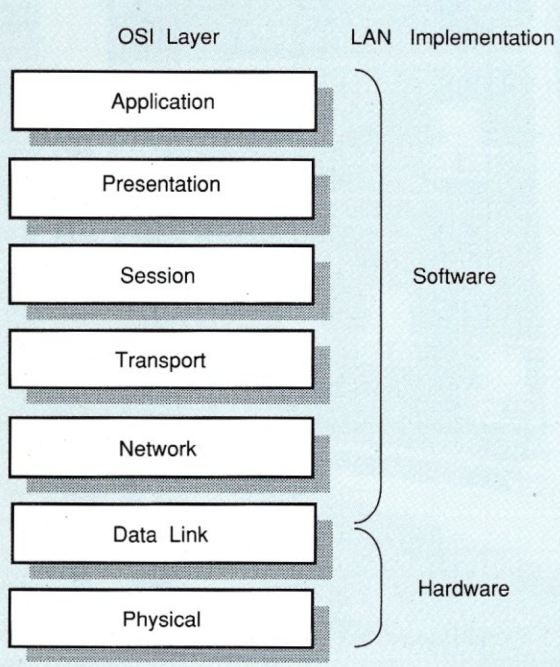
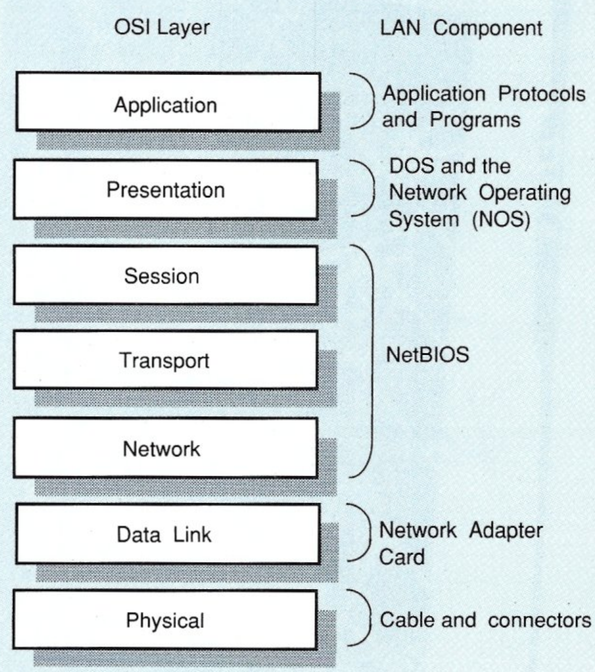
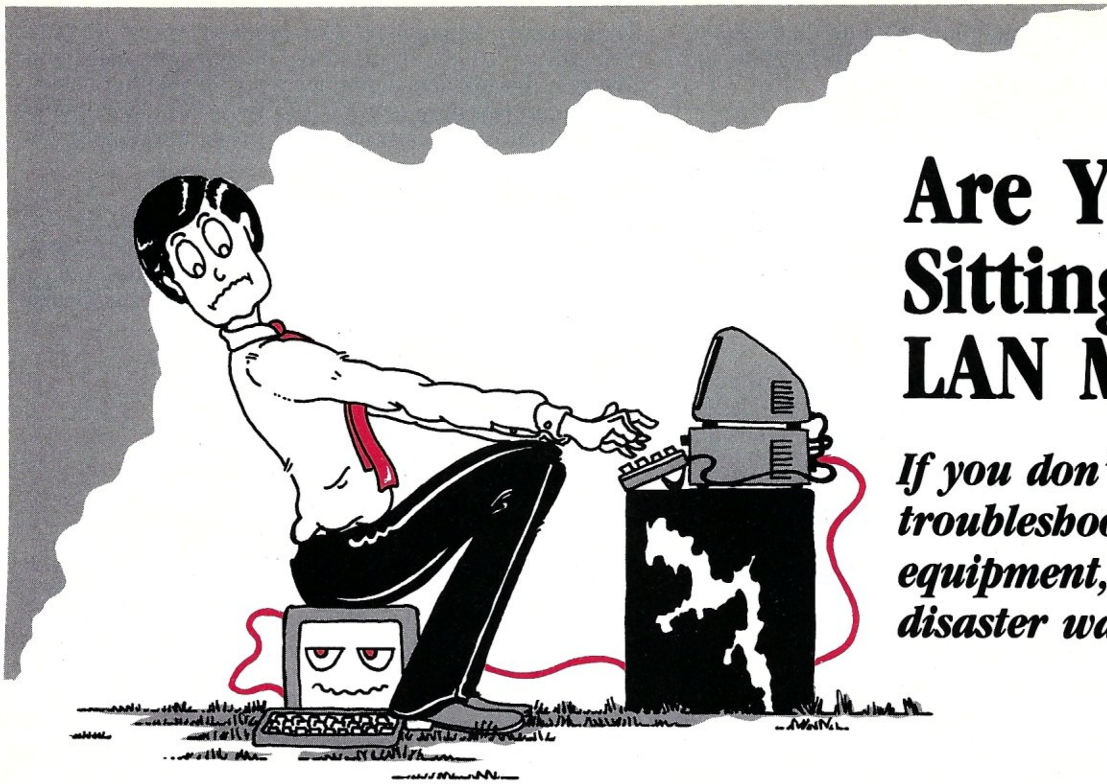


Figure 3. The OSI model applied to LAN components





Are You Sitting on a LAN Mine?

If you don't have the right troubleshooting skills and equipment, your LAN is a disaster waiting to happen!

That's why Data-Tech Institute developed this vital, two-day seminar:

"TROUBLESHOOTING A LOCAL AREA NETWORK"

Finally, the kind of focused training you've been looking for with proven, practical solutions that really work! It's the most effective way to minimize network downtime and maintain peak performance.

You'll learn how to isolate hardware and software failures, apply preventive maintenance procedures and master proven troubleshooting techniques. Plus, find out about the latest test equipment and what you'll need for your "LAN Tool Kit."

For a complete course brochure and outline, call Data-Tech Institute toll free at 1-800/638-5966 or in New Jersey (201) 478-5400 and ask for Jackie or Lisa. Or just fill out and mail or FAX the coupon on right.

And find out how our specialized training helps diffuse potential LAN disasters. Your business is depending on it.

Upcoming Seminar Dates and Locations:

10/11-12	New York, NY
10/13-14	Boston, MA
10/24-25	Los Angeles, CA
10/27-28	Honolulu, HI
11/9-10	Denver, CO
11/15-16	Fairfield, NJ
11/17-18	Detroit, MI
12/1-2	Washington, DC
12/8-9	Pittsburgh, PA
12/15-16	Dallas, TX

Call for additional public dates and private, on-site presentations

YES. ***send me a detailed brochure for your seminar "Troubleshooting a Local Area Network."***

Name__________ Title __________

Company __________

Address__________ State_____ Zip_____

City __________

Telephone (   )__________

Just fill out and mail or FAX.
Our FAX number is 1-201-478-0126.

DATA-TECH INSTITUTE

Lakeview Plaza
P.O. Box 2429, Clifton, New Jersey 07015
1-800-638-5966 • In New Jersey (201) 478-5400

CIRCLE 91 ON READER SERVICE CARD.

so on. NetBIOS is either found in a ROM on the Network Adapter, or emulated in software at the workstation and server.

Layer 6: Implemented in the Network Operating System (NOS) and its associated DOS emulation. Examples include Novell's NetWare, IBM's PC LAN program, 3Com's 3+, Western Digital's ViaNet, Banyan VINES, and others.

Layer 7: An application protocol, and/or an application program. Examples include file transfer and electronic mail applications.

Like the tiered-cake analogy, higher layer protocols provide input for the information fields of the layers below. Figure 4 illustrates this principle for a Transmission Control Protocol/Internet Protocol (TCP/IP) message transmitted on an IEEE 802.3 LAN. The TCP data is at the very core of the frame, encapsulated by the TCP Header, IP Header, and finally the 802.3 Header (Preamble, Start Frame Delimiter, Destination Address, Source Address, and Length fields) and Trailer (Pad and CRC).

The LAN Toolkit

Having looked at the OSI model as it applies to LANs, I will now examine the tools that should be included in a LAN Manager's toolkit—ready to use if a network fault is detected.

Breakout Box (BOB): This box allows the status of the interface leads to be displayed and monitored. Its greatest importance is its ability to reconfigure the interface by opening a path between the interface connectors, and rearranging that path. In this manner, a BOB can be used to quickly configure a null modem cable.

Pulse Trap: Similar in connection to a breakout box, the pulse trap monitors the leads of an interface and visually records any activity (high-to-low or low-to-high transitions) that occurs on any lead. This can be useful in recording extremely fast pulse activity that may not be visibly recognizable.

Bit or Block Error Rate Tester (BERT/BLERT): BERT/BLERT is a digital interface testing device that compares a received signal with a known transmitted signal to determine if any bit or block errors have occurred. A typical BERT/BLERT will operate at various data rates, and will have several canned test messages (such as "The quick brown fox . . ."). It may also include application packages (in ROM) for various Layer 2 and 3 protocols, such as asynchronous, SDLC, X.25, and so on.

Transmission Impairment Measurement System (TIMS): This device is used to measure the analog impairments that may disrupt data communication on telephone lines. Parameters that may be measured by TIMS include: signal-to-noise ratios, line loss, impulse noise, envelope delay, phase jitter, and so on.

Volt-Ohm-Milliammeter (VOM): As the name implies, this instrument measures potential (voltage), resistance (ohms), and current (milliamperes). Typically packaged in a calculator-sized box, the VOM will have a liquid crystal display (LCD), and measure the following ranges:

Volts:	100 microvolts to 1000 volts
Ohms:	0.1 ohms to 20 megaohms
Amperes:	1 microamps to 2000 milliamps

Signal Tracer: This device is an audible tone generator connected to one pair of wires in a cable. The audio signal is traced (or listened to) at the other end with a telephone test set. The primary function of the signal tracer is to determine where twisted pairs go inside the walls of a building when the history of rearrangements, bridged taps, and so on is unknown.

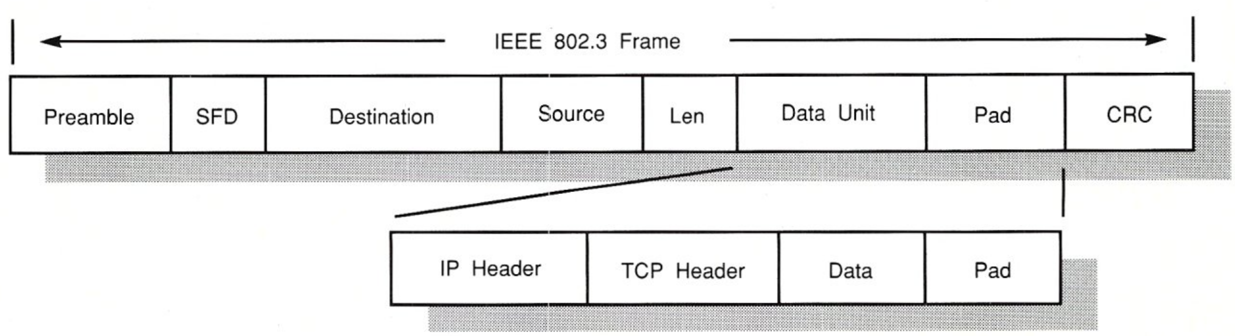
Time Domain Reflectometer (TDR): By sending a short pulse down the cable, the TDR measures the time for the reflection, and calculates the distance to the open or short as a result.

Protocol Analyzers: These hardware/software devices allow the user to determine what is occurring at the bit level of the transmitted signal. Included may be the ability to decode and convert the bit stream into several different formats, for example, wide area network (such as X.25 or T-1) or LAN (such as Ethernet, StarLAN, ARCNET, or Token Ring) protocols.

Oscilloscope: This graphic device displays the signal voltage (vertical axis) per unit time (horizontal axis), and, as such, provides a true representation of analog signals. It can be used to measure the voltage output of RS-232 or RS-422 interfaces, and to analyze the noise component of commercial power sources.

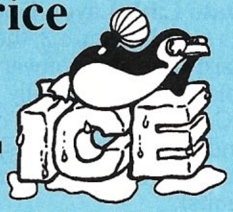
Spectrum Analyzer: These units scan the analog cable spectrum to identify the frequencies of the various carriers (or channels) and their respective guard bands. These carriers are then tested for signal strength, signal-to-noise

Figure 4. Encapsulating the TCP/IP data within an IEEE 802.3 frame



SERIOUS DEBUGGING *at a* REASONABLE PRICE

All the speed and power of a hardware-assisted debugger at a software price



Soft-ICE

Hardware-level break points
REAL-TIME break points on memory locations, memory ranges, execution, I/O ports, hardware and software interrupts. More powerful break points than ANY software-only debugger on the market. Soft-ICE gives you the power of an in-circuit emulator on your desk.

Break out of hung programs
With a keystroke - no external switch necessary. Even with interrupts disabled.

Breaks the 640K barrier
Soft-ICE uses ZERO bytes of memory in the first 1MB of address space. This is especially useful for those subtle bugs that change when the starting address of your code changes. With Soft-ICE your code executes at the same address whether the debugger is loaded or not.

Works with your favorite debugger
Soft-ICE can be used as a stand-alone debugger or it can add its powerful break points to the software debugger you already use. You can continue to use your favorite debugger until you require Soft-ICE. Simply pop up the Soft-ICE window to set powerful real-time break points. When a break point is reached, your debugger will be activated.

Solve tough systems problems too
Soft-ICE is ideal for debugging TSRs, interrupt handlers, self booting programs, DOS loadable device drivers, non-DOS operating systems, and debugging within DOS & BIOS. Soft-ICE is also great for firmware development because Soft-ICE's break points work in ROM.

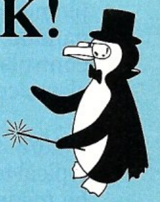
How Soft-ICE Works
Soft-ICE uses the power of the 80386 to surround your program in a virtual machine. This gives you complete control of the DOS environment, while Soft-ICE runs safely in protected mode. Soft-ICE uses 80386 protected mode features, such as paging, I/O privilege level, and break point registers, to provide real-time hardware-level break points.

"Soft-ICE is a product any MS-DOS developer serious enough to own a 386 machine should have."
Dr. Dobb's Journal — May 1988

Both require 80386 AT compatible or IBM PS/2 Model 80. MagicCV requires at least 384K of extended memory. CodeView is a trademark of Microsoft Corporation.

RUN CODEVIEW IN ONLY 8K!

MagicCV



CodeView is a great integrated debugger, but it uses over 200K of conventional memory. MagicCV uses advanced features of the 80386 microprocessor to load CodeView and symbols in extended memory. This allows MagicCV to run CodeView using less than 8K of conventional memory on your 80386 PC.

Don't let 640K be your limit!
If you are closing in on the 640K limit and would like the power of CodeView, MagicCV is for you.

Don't let the debugger hide the bug!
Even if you're not closing in on the 640K limit, running CodeView with MagicCV makes your debugging environment much closer to the end user's program environment. You can use CodeView to locate subtle bugs that only occur when there is plenty of free memory, or those difficult bugs that only occur when your program is running with a couple of TSRs loaded.

How MagicCV works
MagicCV uses the 80386 to create a separate virtual machine for CodeView. MagicCV uses between 4K & 8K of conventional memory as a bridge between the DOS environment and CodeView.

MagicCV is easy to use
If you are a CodeView user, you already know how to use MagicCV too. Just type MCV instead of CV; everything else is automatic.

Save $86

MagicCV with Soft-ICE

Using Soft-ICE with CodeView gives you the features necessary for professional level systems debugging. MagicCV and Soft-ICE can work in concert with CodeView to provide the most powerful debugging platform you will find anywhere.

MagicCV	$199
Soft-ICE	$386

Buy Both and Save $86!

CALL TODAY
(603) 888 - 2386
***or FAX* (603) 888 - 2465**

30 day money-back guarantee
Visa, Master Card and AmEx accepted

NU-MEGA TECHNOLOGIES
P.O. BOX 7607 • NASHUA, NH 03060-7607

CIRCLE 103 ON READER SERVICE CARD.

ratio, and so on, to determine the overall health of the broadband network.

Status Monitor: Used for broadband cables, this monitor incorporates a controller in the head-end along with complementary units at the amplifiers, power supplies, terminations, and so on. Software within the status monitor controls tests and measurements, and responds to preset alarm conditions.

Now that I have addressed the model and looked at typical tools for LAN fault analysis, some correlation must be made among the theoretical (what the model defines as a function), the actual (what characteristics the LAN is exhibiting), and the practical (what tool to pick from the bag to use first).

Physical Layer Testing

The physical layer (cable, connectors, and so on) can be further subdivided into baseband and broadband transmission technologies. Without going into the well-worn controversy as to the superiority of one technology over the other, suffice it to say that baseband transmission with its digital signaling requires a different set of tools than broadband systems, which employ analog transmission. Both systems exhibit similar physical layer problems, that is, the inability of one workstation to access another.

For testing at this level, assume that the peripheral or workstation (PC, terminal, host computer, or whatever) has been tested independently of the network —that is, in a stand-alone mode —and has been found to be functioning properly. This test assures us that the problem is truly a network problem and not a peripheral problem (which is many times incorrectly diagnosed as a network problem).

Baseband Physical Layer Testing

The breakout box, pulse trap, and BERT/BLERT are most frequently used to test interfaces between the LAN and some external device, such as a modem or gateway.

The transmission impairment measurement system (TIMS) is used to analyze analog telephone lines, and is used only when there is reason to believe that the telephone lines connected to the LAN are experiencing difficulties.

The volt-ohm-milliammeter (VOM), signal tracer, and time-domain reflectometer (TDR) can all be used to test faults on copper-based cables (twisted-pair and coaxial). The TDR sends a short pulse down the cable, and receives a reflected pulse from any opens or shorts along the cable. The TDR is most useful for testing long runs of coaxial cable, or for verifying continuity of intrabuilding twisted pairs. Shorter lengths of cable can be tested more easily with a signal generator and tone detector, or with a battery and voltmeter. Finally, an oscilloscope can be used to measure the amplitude and time-domain characteristics of the signal to determine if the transmitter circuitry is operating properly.

Broadband Physical Layer Testing

Because of the analog nature of broadband transmission, not only are these systems more expensive to install, but the diagnostic tools are more complex as well. Most of the test equipment had its origins in the cable TV industry, and has been adapted for use within a broadband LAN environment. A fault in a frequency-division multiplexed broadband system will exhibit a characteristic similar to that in baseband —that is, the inability of individual workstations to communicate. However, the problem is exacerbated because of the additional complexity associated with broadband signal generation. While TDRs can be used to test for cable faults as in the baseband case, the analog nature of the transmission requires a spectrum analyzer or status monitor in order to determine if any network faults are the result of frequency misalignments or other frequency-based aberrations.

Data Link Layer Testing

As seen in Figure 2, the data link layer encompasses both hardware and software components. Network faults at this layer are typically exhibited by either a failure of the PC or workstation to connect to the network (hardware and/or software faults) or a network address that cannot be recognized or acknowledged (software or firmware faults).

To eliminate the possibility of hardware faults, four simple steps can be used:

1. Check the PC for stand-alone operation using the diagnostic utilities that came with the PC.
2. Check the cable connection into the Network Adapter. Also make sure that the gold fingers on the Network Adapter are clean (a pencil eraser works well for this) and firmly seated into the PC bus connector. Check any switch settings or jumpers on the board for proper configuration.
3. Swap out the Network Adapter Card with a known good spare.
4. Check for any interactive problems between the Network Adapter and any other add-in boards, such as video cards. Oddly enough, the easiest way to isolate and correct the problem is to take any newly added boards out of the PC and add them back one at a time. Another technique is to use a PC diagnostic program to determine if any interrupt request (IRQ) or direct-memory access (DMA) channels have been selected by both the add-in boards and Network Adapter cards. Consult the Network Adapter documentation to avoid these interactive conflicts.

However, all data link faults do not need to be hardware-related. As discussed previously, the data link layer's principal functions are to frame the data, add source and destination addressing, and provide for a point-to-point error check, or CRC. Since these functions are principally software- (or firmware-) based, the appropriate diagnostic tool for these faults is the protocol analyzer.

The frame formats of Ethernet, Token Ring, ARCNET, StarLAN, and others are all different, and the appropriate protocol analyzer with its associated software is required in order to diagnose these faults properly. Many protocol analyzers are based in a PC, such as a Compaq Portable II, and attach to the LAN in either an active or passive mode, as shown in Figure 5. In the passive mode, the analyzer captures (or eavesdrops) on data frames passing by on the bus, and checks each frame for the proper length, correct CRC, and so on. The information field of the frame may not be examined at this layer, but it is generally reserved for analysis by a higher-layer protocol interpreter, such as TCP/IP, NetBIOS, or XNS. In the active mode, the protocol analyzer can generate (or simulate) a load on the network via broadcast messages to all modes, or a specific pattern to one particular mode.

Higher Layer Testing

Recall from Figure 2, that Layers 3 through 7 encompass software-only functions. Therefore, the principal diagnostic tool for these layers is again the protocol analyzer. A second function of processing is required, however, in order to examine the contents of the data link layer frame.

As we saw in Figure 4, higher layers (3 through 7) are encapsulated within the data link layer information field, and a specific protocol interpreter is required to make sense of this data. Typical protocol interpreters are software packages that are purchased separately, and include options for TCP/IP, NetBIOS, XNS, Novell NetWare (IPX), DECnet, and APPC. Difficulties in communicating with a gateway can also be addressed in a similar manner by using a protocol analyzer on the output (that is, non-network) side of the gateway, as shown in Figure 5.

Protocol analysis at these layers typically uncovers incorrect port addresses, or higher-layer data field lengths that are either too long or too short.

Network Operating System Diagnostics

Layer 6 encompasses DOS and the Network Operating System (NOS). While a protocol analyzer with the proper protocol interpreter can certainly be used for testing at this layer, most operating systems include some rudimentary diagnostic functions within their own code to test for proper node address name and password assignments, correct read/write permission modes, and typical statistics on the operation of the network as a whole (number of collisions, traffic counts, and so on). Since most of these functions are "built-in" and generally very easy to use, it behooves the network administrator to become familiar with these capabilities and check them on a routine basis.

Application Layer Diagnostics

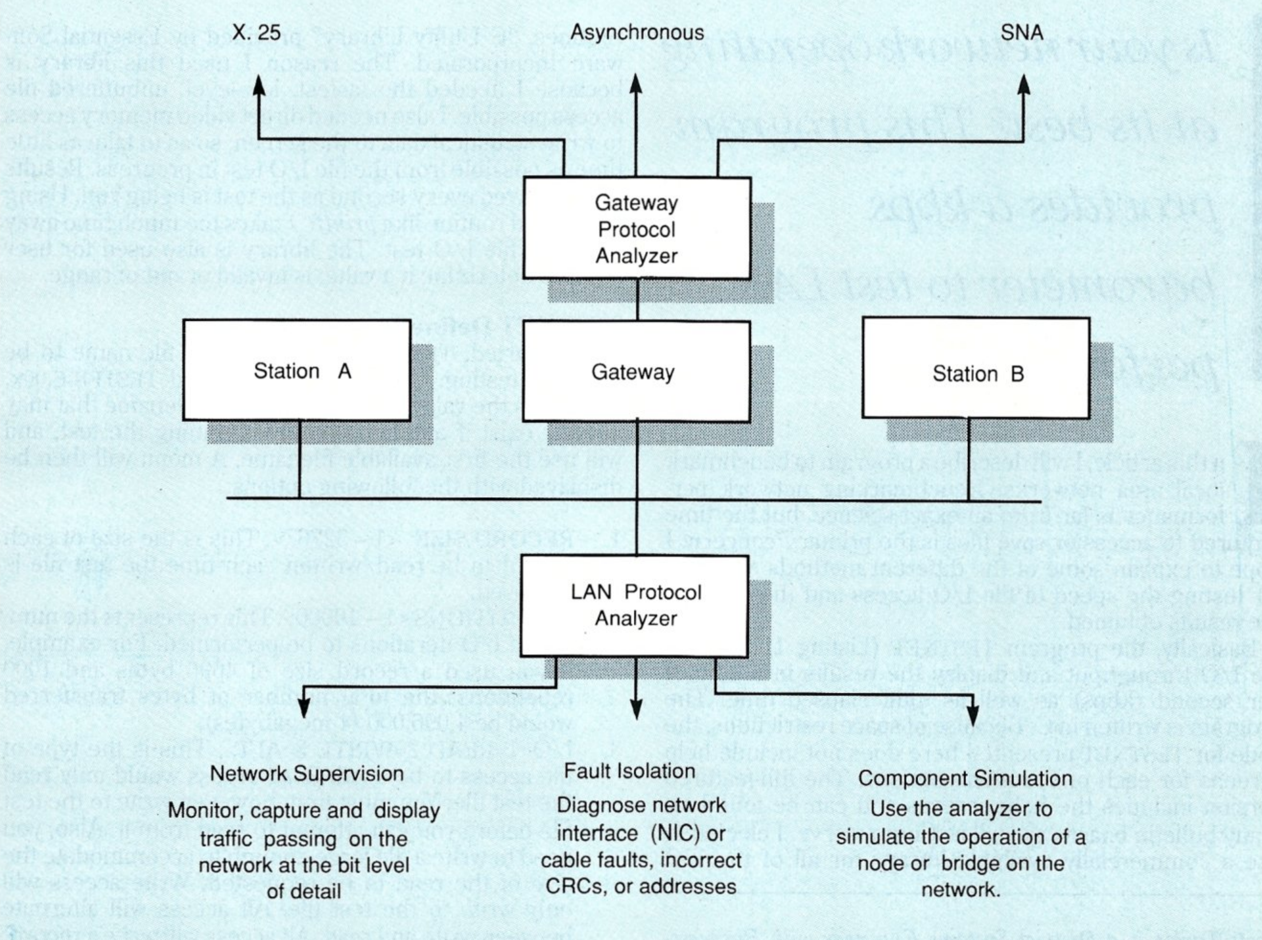
Layer 7 includes any application protocol that may be required for file transfer, message handling, or database access. The application layer is fed from the application program data—for example, the contents of the electronic mail message in a message-handling application. Troubleshooting tools for this layer are the protocol analyzer with the appropriate protocol interpreters, or the documentation that accompanies the specific application program. This is one area where the best tool is a cool head, because no amount of elaborate test equipment can make up for an incorrect printer assignment (LPT1 instead of LPT2, for example) within a specific application program.

Summary

I have addressed the seven-layer OSI reference model from both theoretical and LAN application-oriented views, and also looked at several troubleshooting tools for LAN fault analysis. The best tool, however, is knowledge. A LAN administrator who understands both the operation of the tools and the functions of each component of the LAN can be sure to reach for the right tool to isolate the network fault currently under investigation. □

Did you find this article particularly useful? Circle number 3 on the reader service card.

Figure 5. Protocol analyzer testing configurations



Benchmarking LAN Performance

by Scott Taylor

Is your network operating at its best? This program provides a kbps barometer to test LAN performance.

In this article, I will describe a program to benchmark local area networks. Benchmarking network performance is far from an exact science, but the time required to access or save files is the primary concern. I hope to explain some of the different methods available for testing the speed of file I/O access and the value of the results obtained.

Basically, the program TESTNET (Listing 1) will test file I/O throughput and display the results in kilobytes per second (kbps) as well as total elapsed time. The program is written in C. Because of space restrictions, the code for TESTNET presented here does not include help screens for each of the menu options. The full-featured version includes the help screens and can be found on many bulletin boards as well as Compuserve. I elected to use a commercially available library for all of the I/O routines, "C Utility Library" provided by Essential Software Incorporated. The reason I used this library is because I needed the fastest, low-level, unbuffered file access possible. I also needed direct video memory access to write statistical data to the screen, so as to take as little time as possible from the file I/O test in progress. Results are displayed every second as the test is being run. Using a standard routine like *printf( )* takes too much time away from the file I/O test. The library is also used for user input to determine if a value is invalid or out of range.

Scott Taylor is a District Systems Engineer with Businessland Incorporated in Washington D.C. He has engineered and maintains over 150 local area network installations.

TESTNET Defined

When started, TESTNET first creates a file name to be used for testing. It creates a file named TESTFILE.00x, where *x* is the value 0 – 9. It looks for a filename that may already exist if another station is running the test, and will use the first available filename. A menu will then be displayed with the following options:

1. RECORD SIZE <1 – 32767>. This is the size of each record to be read/written each time the test file is accessed.
2. REPETITIONS <1 – 10000>. This represents the number of I/O iterations to be performed. For example, if you used a record size of 4096 bytes and 1000 repetitions, the total number of bytes transferred would be 4,096,000 (4 megabytes).
3. I/O<1=READ 2=WRITE 3=ALT>. This is the type of file access to be used. Read access would only read the test file. You must first, however, write to the test file before you can attempt to read from it. Also, you need to write a file large enough to accommodate the size of the read to be requested. Write access will only write to the test file. Alt access will alternate between write and read. Alt access will write a record, seek back to the beginning where that write took place, then read the record back. Then it will advance

Listing 1. TESTNET.C

```
/* Include File That Defines Colors For CGA */

#include <color.h>

                                 /* Define values for following keys */
#define F1_KEY 59                   /* F1 Key */
#define F2_KEY 60                   /* F2 Key */
#define RETURN 13                   /* RETURN Key */
#define ESC_KEY 27                  /* ESCAPE Key */
#define UP_ARROW 72                 /* UP ARROW Key */
#define DN_ARROW 80                 /* DOWN ARROW Key */
#define NUMLOCK_UP_ARROW 56         /* NUMLOCK+UP ARROW Key */
#define NUMLOCK_DN_ARROW 50         /* NUMLOCK+DOWN ARROW Key */
#define READ_WRITE 2                /* File Access Method */
#define READ 1                      /* I/O READ */
#define WRITE 2                     /* I/O WRITE */
#define ALTERNATE 3                 /* I/O ALTERNATE */
#define SEQUENTIAL 1                /* SEQUENTIAL File Acces */
#define OVERLAY 2                   /* OVERLAY File Access */
#define INCREMENT 1                 /* INCREMENT Record Size */
#define DECREMENT 2                 /* DECREMENT Record Size */
#define RANDOM 3                    /* RANDOM Record Size */
#define STANDARD 4                  /* STANDARD Record Size */
#define MULTIUSER_START 999         /* Station Starting MultiUser */

                              /* Set following values */
double kb;                    /* Total KiloBytes */
double sec;                   /* Total Elapsed Seconds   */
int io_type;                  /* IO TYPE (Read/Write/Alt) */
int repetitions;              /* REPETITIONS (# Of Records To Write) */
int record_style;             /* RECORD STYLE (Inc/Dec/Rand/Std) */
int file_type;                /* FILE TYPE (Sequential/Overlay) */
int num;                      /* Incrementing & Decrementing */
int val;                      /* Value For */
int input_value;              /* User Input */
int hlp;                      /* Help Screens & Responses */
int keystroke;                /* Menu Keystrokes */
int new_menu_bar=1;           /* New Menu Bar Position (1=Beginning) */
int old_menu_bar;             /* Old Menu Bar Position */
int video_screen[2048];       /* Buffer For Saving Video Screen   */
int mode;                     /* Value For */
int color2;                   /* Standard Help Text Color */
int color3;                   /* HiLight Text Color */
int color4;                   /* Status Line Text Color */
int test;                     /* Testing File I/O Errors */
int chk_write;                /* Checking Write Before Read */
unsigned handle;              /* File Handle Of Test File */
unsigned handle2;             /* File Handle Of MultiUser File */
unsigned bytes;               /* # Of I/O Bytes To Transfer */
unsigned actual;              /* Actual # Of I/O Bytes Transfered */
unsigned count;               /* # Of REPETITIONS Executed */
unsigned record_size;         /* The Record Size To Be Used */
unsigned rnd;                 /* Random Record Sizes To Be Used */
unsigned temp;                /* Temporary Buffer (Loops, Etc...) */
long actual_count;            /* Total # Of Bytes Transfered */
long loc;                     /* LOCATION Of Previous File Position */
long loc2;                    /* LOCATION Of Current File Position */
long ref;                     /* Value For REFERENCE File Position */
long time1;                   /* Current Elapsed Clock Timer Tics */
long time2;                   /* Total Clock Tics To Convert To SEC */
long time3;                   /* Value For Real Time Clock Tics */
long timer();                 /* Definition For Timer Routine */
char buffer[20480];       /* Buffer For Data To Write To Test File */
char filename[13] = {"TESTFILE.000\0"}; /*Default Test File Name*/

MAIN() {

qvideo();              /* Routine For Direct Screen Memory Access */
opening_screen();      /* Displays Opening Credits Screen */
set_attributes();      /* Screen Attributes For CGA Or MONO */
input_value=0;

clscolor(LTWHITE,BLUE);     /* Clear Screen Light White On Blue */
wborder(2,2,17,78,14,BLUE,1);     /* Display Main Window Border */
wborder(3,4,5,76,RED,BLUE,0);     /* Display Title Window Border */
wborder(6,4,13,43,color2,WHITE,0);/* Display Menu Window Border */
curlocat(4,5);
colrprts("    TESTNET.EXE v1.1 - NETWORK FILE I/O TEST      ",0,7);
parameters_window();                /* Display Menu Screen Options */
start_file();              /* Determine & Create Test File For I/O */

start:                              /* Start Location For Testing */

actual_count = ref = 0L;/* Set Total Bytes & File Position To 0 */
curlocat(15,6);
colrprts("                                          ",LTCYAN,BLUE);
old_menu_bar=new_menu_bar;/*Set Old Menu Bar To Current Position*/
keystroke=new_menu_bar=1;  /* Set Keystroke & Menu Bar Pos To 1 */
instruct();                          /* Display Menu Instructions */
select();                 /* Display Menu Bar To Correct Position */
while (keystroke != F1_KEY && keystroke != F2_KEY) {
     getkey(&keystroke);        /* Get User Keystroke Input */
     if (keystroke == ESC_KEY)   /* Test For ESCAPE Key,   */
          quit();             /* If So Quit */
     else if (keystroke == RETURN)     /* Test For RETURN Key, */
          entries(new_menu_bar); /* If So Get Value */
     else if (keystroke == NUMLOCK_UP_ARROW || keystroke == UP_
       ARROW) up_figure(); /* If UP ARROW, Set New Menu Bar Pos*/
     else if (keystroke == NUMLOCK_DN_ARROW || keystroke == DN_
       ARROW) down_figure();/*If Dn ARROW, Set New Menu Bar Pos*/
     else
          shrtbeep();/* If Wrong Keystroke Beep (error) */
}

/* Test To See If Acceptable Values Have Been Given For All
               Parameters ! If Not Then Return To Beginning */
if (record_size <= 0 || repetitions <= 0 || io_type <= 0 ||
            file_type <= 0 || record_style <= 0) {
          curlocat(15,6);
          colrprts("YOU MUST GIVE A VALUE FOR ALL PARAMETERS !
                      Press Any Key...",6,9);
          longbeep();    /* BEEP Indicating An Error Has Occured */
        getkey(&keystroke);  /* Get Any Keystroke To Continue */
          goto start;              /* Return To Start Location */
}

/* Check If User Requested MultiUser Operation. If So Then Enter
               MultiUser. */

if (keystroke == F2_KEY) {
     multiuser();
     if (keystroke == ESC_KEY) /* If User ESCAPES From MultiUser*/
               goto start;     /* Mode Return To START Location */
}
clr_help();                       /* Clear Help Screen Area Of Text */
rnd = record_size;     /* Set Random Counter To Record Size Val */

/* Test If INCREMENT Was Selected. If So Check If REPETITIONS Can
   Be Divided By RECOED_SIZE, If It Cannot Then Maintain Same
   RECORD_SIZE. If It Can Be Divided Then Set NUM To Value To Be
   Incremented. */

if (record_style == INCREMENT) {
          if ( record_size < repetitions)
                    num=0;
          else
                    if (io_type == ALTERNATE)
                              num=(record_size/repetitions)/2;
                    else {
                              num=record_size/repetitions;
                              record_size=1;
                         }
}

/* Test If DECREMENT Was Selected. If So Check If REPETITIONS Can
   Be Divided By RECOED_SIZE, If It Cannot Then Maintain Same
   RECORD_SIZE. If It Can Be Divided Then Set NUM To Value To Be
   Decremented. */
if (record_style == DECREMENT) {
          if ( record_size < repetitions)
                    num=0;
          else
                    if (io_type == ALTERNATE)
                              num=(record_size/repetitions)/2;
                    else
                              num=record_size/repetitions;
}

chk_write=chk_write+io_type;
if (chk_write == 1) {     /* If READ Was Selected Before WRITE 8/
          curlocat(15,13); /* Then ERROR */
          colrprts("MUST WRITE TO FILE BEFORE ATTEMPTING TO READ",
          BROWN,LTBLUE);
          goto error;
}

test = openfil(filename,READ_WRITE,&handle);      /* Open File */
if (test != 0) {                /* Check For Error Opening File */

           curlocat(15,20);
           colrprts("ERROR ACCESSING TEST FILE !",BROWN,LTBLUE);
           goto error;
 }

 curlocat(8,59);
 colrprtf(0,LTGREEN,BLUE,"-FILENAME-");
 curlocat(9,58);
 colrprtf(0,LTGREEN,BLUE,"%s",filename); /* Display Test Filename
                          Being Used */

 temp=io_type;             /* Temporarily Save IO_TYPE Selected */
 if (temp == ALTERNATE)    /* If ALTERNATE Set Flag To READ */
           io_type=READ;

 time2=time1=0L;
 time3=timer();
```

. . . listing continues

Listing 1. *(continued from page 27)*

```
for (count = 1;count <= repetitions;count++) { /*Set Count Loop*/
     if (temp == ALTERNATE) {   /* If ALTERNATE Was Selected */
          if (io_type == READ) {  /* Switch Between READ & WRITE */
                 io_type=WRITE;   /* And Then Subtract COUNT To */
                 count--;         /* Accomadate READ & WRITE */
                 seekfil(handle,0,ref,&loc); /* Seek To End Pos */
          }
          else if (io_type == WRITE) {
                    io_type=READ;    /* Switch Between READ & WRITE */
                    seekfil(handle,0,loc,&loc2); /*Seek To Prev. Pos*/
          }
     }
     if (io_type == READ)
         readfil(handle,record_size,buffer,&actual);/*READ Buffer*/

     else {
         writefil(handle,record_size,buffer,&actual);/*WRITE Bufr*/
         ref=ref+actual; /* Set REFERENCE LOcation To End Of File /
     }

     if  (file_type == OVERLAY) /*If OVERLAY Then Repeated Seek*/
         seekfil(handle,0,0L,&ref); /* To Biginning Of File  */

     if (actual < record_size) {      /* Bytes Transfered Less */
         curlocat(15,10); /* Than RECORD_SIZE then Display Error */
         colrprts("DISK I/O ERROR - (POSSIBLY OUT OF DISK SPACE)",
              BROWN,LTBLUE);
                  goto error;
     }
     actual_count=actual_count+actual;  /* Count Total Bytes */
     timel=timel+(time3=timer());       /* Count Time Elapsed */
     if(timel >= 18) {      /* If Greater Than 18 Clock Tics */
                  results();  /* (1 second) Then Display Results */
     }

     if (record_style == DECREMENT) /* If DECREMENT Decrease */
         record_size=record_size-num; /*RECORD_SIZE By NUM */

     else if (record_style == INCREMENT)/*If INCREMENT Increase*/
         record_size=record_size+num; /*RECORD_SIZE By NUM */

     else if (record_style == RANDOM) { /*If RANDOM Set RECORD_*/
         record_size = randnum(1,rnd); /* SIZE To Random (RND) */

         if (io_type == WRITE && temp == ALTERNATE && record_size
          > actual) record_size = actual; /*If ALTERNATE & */
         }               /* RANDOM Set RECORD_SIZE To Last WRITE */
}                       /* Value For READing */

timel=timel+(time3=timer()); /* Count Total Time Elapsed */
record_size=rnd;             /* Set RECORD_SIZE To Random Value */
io_type=temp;                /* Reset IO_TYPE To Original Value */
count--;                     /* Reset COUNT To Actual Value */
results();                  /* Display Total Statistics Results */

curlocat(20,2);              /* Display Total Elapsed Seconds */
colrprtf(0,LTCYAN,BLUE,"Total seconds = %3.1f",sec);

error:                          /* Location For Error Handling */
if (keystroke == 1) {           /* If In MultiUser Mode */
        closefil(handle2);      /* Close MultiUser Queue File */
        unlink("\\TESTFILE.QUE"); /* And Delete The Queue File */
}

closefil(handle);                /* Close Test Filename */
curlocat(21,2);

/* Get YES/NO Response. If NO Exit Program. If YES Continue */
colrprts("Continue Testing ? (Y/N)",LTCYAN,BLUE);
if (getyesno(1) == 1) {
          curlocat(20,2);
          colrprts("                          ",LTCYAN,BLUE);
          curlocat(21,2);
          colrprts("                          ",LTCYAN,BLUE);
          goto start;
}
quit();
}

/* Get User Input Value & Tests For Min-Max Ranges */
entries(hlp) {
     input_value=0;
     switch(hlp) {
          case 1:
          while (input_value < 1) {
            curlocat(7,35);
            colrprts("     ",0,7);
            curlocat(7,35);
            input_value = getunint(5,9,0,1,2,&record_size,1,1,32767);
            if (input_value < 1)
                 longbeep();
          }
          case 2:
          while (input_value < 1) {
            curlocat(8,35);
            colrprts("     ",0,7);
            curlocat(8,35);
          input_value = getint(5,9,0,1,2,&repetitions,1,1,10000);
          if (input_value < 1)
                  longbeep();
        }
          case 3:
          while (input_value < 1) {
            curlocat(9,35);
            colrprts(" ",0,7);
            curlocat(9,35);
            input_value = getint(1,9,0,1,2,&io_type,1,1,3);
            if (input_value < 1)
                    longbeep();
        }
          case 4:
          while (input_value < 1) {
            curlocat(10,35);
            colrprts(" ",0,7);
            curlocat(10,35);
            input_value = getint(1,9,0,1,2,&file_type,1,1,4);
            if (input_value < 1)
                    longbeep();
          }
          case 5:
          while (input_value < 1) {
            curlocat(11,35);
            colrprts(" ",0,7);
          curlocat(11,35);
            input_value = getint(1,9,0,1,2,&record_style,1,1,4);
            if (input_value < 1)
                    longbeep();
            }
   }
          instruct();    /* Display Menu Keystroke Options */
          down_figure(); /* Move Menu Bar Down One Pos. */
}

/* Position Menu Bar To Its New Location And Resets
   Old Location Of Menu Bar To Its Original Screen Attributes */
select() {
          paint(6+old_menu_bar,5,6+old_menu_bar,32,BLACK,WHITE);
          paint(6+new_menu_bar,5,6+new_menu_bar,32,LTWHITE,LTRED);
}

/* This Section Clears Help Screen Area */
clr_help() {
          for (temp=1;temp<10;temp++) {
                    curlocat(6+temp,47);
                    colrprts("                          ",LTCYAN,BLUE);
          }
}

/* Move Menu Bar Up One Position. It Will Roll Down To
   Bottom Of Menu If Already At Top */
up_figure() {
          old_menu_bar=new_menu_bar;
          new_menu_bar--;
          if (new_menu_bar == 0)
                   new_menu_bar=5;
         select();               /* Paint Menu Bar To Screen */
}

/* Move Menu Bar Down One Position. Roll Up To
   Top Of Menu If Already At Bottom */
down_figure() {
          old_menu_bar=new_menu_bar;
          new_menu_bar++;
          if(new_menu_bar == 6)
                    new_menu_bar=1;
          select();                /* Paint Menu Bar To Screen */
}

/* Display Menu Options Instructions */
instruct() {
     clr_help();
   curlocat(7,47);
     colrprts("<ESC> = Exit to system",LTGREEN,BLUE);
   curlocat(8,47);
     colrprts("<F1> = Start Testing",LTGREEN,BLUE);
   curlocat(9,47);
     colrprts("<F2> = Start Multi-Station Test",LTGREEN,BLUE);
   curlocat(10,47);
     colrprtf(0,LTGREEN,BLUE,"<%c> & <%c> = Select Menu Option",
          24,25);
   curlocat(11,47);
     colrprts("<ENTER> = Enter Data For Option",LTGREEN,BLUE);
   curlocat(25,1);
}

/* Prepare To Exit Program. First Delete Test File Created,
     Then Set Screen Attributes To White on Black */

quit() {
```

to the end of the file and continue on in the same manner until all repetitions are completed.

4. FILE <1=SEQUENTIAL 2=OVL>. This represents how the records will be treated within the test file. Sequential file access means that each record will be read, or written, at the end of each previously accessed record (appended). OVL or overlaid file access means that each record will be read, or written, at the beginning of the file using the same space.
5. RECORD MANIPULATION STYLE <1=INC 2=DEC 3=RAND 4=STD>. This determines the size of the record to be used for each I/O iteration. The INC option will increment the size of each record for every iteration, starting at 1 byte and increasing to a maximum of the record size given in the menu option. If the record size option given in the menu is less than the number of repetitions, then the size of each record will remain the same. The DEC option will decrement the size of each record for every iteration, starting at the value given in the record size option of the menu and decreasing down to 1 byte. As stated above, if the record size option given in the menu is less than the number of repetitions, then the size of each record will remain the same. The RAND option will use random record size for each iteration. This value will range from 1 byte to a maximum of the record size value given in the menu. The STD option will maintain a standard record size. This would be the value given in the record size option of the menu.

> *This program will show the total elapsed time as well as the average throughput in kilobytes per second transferred to and from the disk.*

Initializing the Test

There are two ways in which to initiate the test. The first option is the stand-alone method activated by pressing the F1 key. This means that the station under test will operate independently from any other stations. The second option is the multiuser method, activated by pressing the F2 key on each station to be used for testing. Once each station has been activated for the multiuser test mode, you simply press the F1 key at any one of those nodes and all stations will commence testing. All the stations to be run in the multiuser mode must have access to same root directory because all stations will constantly look for a file named TESTFILE.QUE in the root directory. This file is created by the station that starts the multiuser testing (pressing the F1 key), and is deleted when testing has completed. When this file is created, all stations will find it and thus will start testing.

The valuable results of this program will show the total

```
          unlink(filename);
          clscolor(WHITE,BLACK);
          exit();
  }

  /* Check For Type Of Monitor Being Used. Then Set Up Colors For
       Certain Window Attributes (For Readability And Contrast) */

  set_attributes() {
          if ( ifmono() ) {
                  color2=BLACK;
                  color3=LTWHITE;
                  color4=WHITE;
          }
          else {
                  color2=LTCYAN;
                  color3=YELLOW;
                  color4=LTGREEN;
          }
  }

  /* MultiUser (MultiStation) test routine */
  multiuser() {
       savwindo(0,0,24,79,video_screen);/*Save Video Pg To Buffer*/
       for(temp=8;temp<22;temp++) {      /*Clr Window Area Of Text*/
            curlocat(temp,15);
            colrprts("                                          ",15,6);
       }
       /* Write MultiUser Help Text To Window & Await Response */
       wborder(8,15,22,65,15,4,1);
       curlocat(10,22);
       colrprts("MMM MULTI-STATION TESTING MODE  MMM",LTCYAN,BROWN);
       curlocat(13,21);
       colrprts("Mode allows mulitple stations",LTWHITE,BROWN);
       curlocat(14,21);
       colrprts("to run I/O tests at same time.",LTWHITE,BROWN);
       curlocat(16,21);
       colrprts("Stations participating in test",LTWHITE,BROWN);
       curlocat(17,21);
       colrprts("must be set to this mode (pressing",LTWHITE,BROWN);
       curlocat(18,21);
       colrprts("<F2> from main screen).  To begin",LTWHITE,BROWN);
       curlocat(19,21);
       colrprts("test, press <F1> from last work",LTWHITE,BROWN);
       curlocat(20,21);
       colrprts("station activated.  Stations will",LTWHITE,BROWN);
       curlocat(21,21);
       colrprts("then start test.  <ESC> TO ABORT !",LTWHITE,BROWN);
       for(temp=1;temp<5;temp++) {       /* Beep 4 Times As Alert */
            for(hlp=1;hlp<1000;hlp++) {   /* Run Delay Loop */
            }
           shrtbeep();
     }

 /* Repeatedly Test For Existance Of TESTFILE.QUE. If It Exists
 Set Keystroke To MULTIUSER_START To Indicate This Station Did
 Not Initiate Test, Then Begin Testing. If It Does Not Exist Then
 Check For F1_KEY (To Create TESTFILE.QUE & Begin Tests), Or Check
 For ESC_KEY (To Abort Testing). */

 while( keystroke != F1_KEY && keystroke != ESC_KEY && keystroke !=
           MULTIUSER_START) {
         inkey(&keystroke);
         if (filexist("\\TESTFILE.QUE") != 0)
               keystroke=MULTIUSER_START;
         }
         if (keystroke == F1_KEY) {
                 creatfil("\\TESTFILE.QUE",32,&handle2);
                 keystroke=1;
         }

         rstwindo(0,0,24,79,video_screen);
}

/* Create Unique Filename For Testing. Check For Existence Of
   Filename Called TESTFILE.00x, Where x Is digit 0-9.
   Create First One That Does Not Already Exist */
start_file() {
     for( temp=49;temp<59;temp++) {
          if( temp == 58 ) {
                 clscolor(WHITE,BLACK);
                 curlocat(21,2);
               colrprts("CANNOT CREATE FILE FOR TESTING,
                   ABORTING !",WHITE,BLACK);
               exit();
        }
         filename[11]=temp;
         if( filexist(filename)==0 ) {
                temp=125;
                creatfil(filename,32,&handle);
        }
     }
}
```

. . . listing continues

elapsed time in seconds as well as the average throughput in kilobytes per second transferred to, or from, the disk. Using a sequential file access method primarily demonstrates the server's disk and/or caching performance. Running a sequential read or write test on a file that is smaller than the total cache memory available in the server will reflect more on the type of network adapter being used. To do this effectively, you should use a high-speed 286- or 386-type workstation and server. In my experience I have found that when using an adapter such as a Token-ring type card, there is negligible difference in throughput between using an 8088-type station and using an 80386-based station. On the other hand, if you use an adapter such as an Ethernet card, there can be as much as a 400 percent increase in average throughput between using an 8088-based machine and using an 80386-based machine. Using an overlaid file access method tests local cache techniques because the same record is read/written over and over again in local memory. If no local caching exists, or you use a record size larger than the amount of local caching available, then you are testing the time required for the server and disk to repeatedly seek back to the beginning of the file.

> *Benchmarking network performance is far from an exact science, but the time required to access or save files is the primary concern.*

Larger record sizes are handled more efficiently by most operating systems. Using larger record sizes (above 4K) will demonstrate maximum throughput ability, it also will test the server's capacity to handle a vast quantity of data. Smaller record sizes show the operating systems capability to deal with small record blocks. A good example would be to run tests from a workstation's local "C:" drive. Then run the tests again from that same workstation but to a network drive. Usually you will find a significant increase in performance. You might also try running the tests with the DOS verify option on and compare the results.

Here are some good examples, especially for comparing servers (disk, memory, and CPU speed). This is especially useful when using an operating system like Novell's NetWare, since the server displays activity statistics. You should use a high-speed test station and a very fast I/O network adapter card to reduce any impeding bottleneck in those areas. To compare hard disks, generate a file size using TESTNET that is larger than the total memory available in the server. This way the file size is too large to fit in cache memory and tests raw disk I/O. Then you can compare throughput times displayed by TESTNET. To compare server CPUs and cache memory speed, write a test file size that is smaller than the available cache memory installed in the server. Then read the file. This will test the raw CPU and cache memory performance. Then you can compare the results. If you use an operating system such as NetWare, while the test is in progress you can monitor the server console for peak "I/O PENDING" and peak "% UTILIZATION." This way you can compare the server workload statistics with the relative time it takes for disk access. □

Did you find this article particularly useful? Circle number 4 on the reader service card.

Product Information

C Utility Library

Essential Software, Inc.
P.O. Box 1003
Maplewood, NJ 07040
(201) 762-6965
Circle reader service #251.

Listing 1. *(continued from page 29)*

```
results() {
     time2=time2+time1;             /* TIME2 = Total Clock Tics */
     time1=0L;                 /* Reset TIME1 For Next COUNT Loop */
     sec=time2/18.2;           /* Set SEC To Total Elapsed Seconds */
     kb=(actual_count/sec)/1024; /* Figure Total KB Transfered */
     curlocat(15,6);
     colrprtf(0,LTCYAN,BLUE,"TIME = %3.0f  I/O COUNT = %d   TOTAL
         BYTES = %ld   KB/SEC = %3.1f",sec,count,actual_count,kb);
}

/* Display Text Options Inside Menu Window */
parameters_window() {
     curlocat(7,5);
     colrprts("RECORD SIZE <1 - 32767>   ] 512     ",BLACK,WHITE);
     record_size=512;          /* Set Record Size Default To 512 */
     curlocat(8,5);
     colrprts("REPETITIONS <1 - 10000>   ] 1000    ",BLACK,WHITE);
     repetitions=1000;         /* Set REPETITIONS Default To 1000 */
     curlocat(9,5);
     colrprts("I/O <1=READ 2=WRITE 3=ALT> ] 2      ",BLACK,WHITE);
     io_type=WRITE;            /* Set IO TYPE Default To WRITE */
     curlocat(10,5);
     colrprts("FILE <1=SEQUENTIAL 2=OVL>  ] 1      ",BLACK,WHITE);
     file_type=SEQUENTIAL;  /* FILE TYPE Default = SEQUENTIAL */
     curlocat(11,5);
     colrprts("RECORD MANIPULATION STYLE  ] 4      ",BLACK,WHITE);
     record_style=STANDARD; /* RECORD STYLY Default = STANDARD */
     curlocat(12,5);
     colrprts("<1=INC 2=DEC 3=RAND 4=STD> ]        ",BLACK,WHITE);
}

/* This Section Displays The Opening Credit Screen */
opening_screen() {
     clscolor(LTWHITE,BLACK);
     colrprtf(1,LTWHITE,BLACK,"\n\n\n\n        TESTNET.EXE v1.1");
     colrprtf(1,LTWHITE,BLACK,"\n\n              By SCOTT TAYLOR");
     syspause(0,0,4,0);    /* Pause For 4 Seconds Then Continue */
}
```

End Listing 1

Due to space limitations, the listings that appear with this article have been abbreviated from the original source material. The complete version is available on MS-DOS disk. To order, send $14.95 to Micro/Systems Journal, *501 Galveston Drive, Redwood City, CA 94063; or call Tim at (415) 366-3600. Please specify the issue number. Source code is also available on CompuServe; type GO DDJFORUM and check the data library, line 14.*

LAN Analysis:
Choosing the Right System

A product sampler of four commercial approaches to network monitoring.

On the preceding pages, Scott Taylor provides a do-it-yourself approach to measuring LAN performance. However, comprehensive LAN analysis is a complicated task and there are a growing number of commercial LAN analysis tools that offer different approaches to network analysis. These products come in a wide range of price tags and configurations. Some are software analyzers that perform simple error checking. Others are hardware solutions that can be used as portable fluoroscopes to pinpoint problems in an ailing network. Still others are hardware/software combinations that can be installed in a dedicated PC workstation to act as traffic cops that monitor network traffic and report on activity.

To provide a better understanding of the analysis options that are available, we contacted four LAN analyzer vendors and asked them about their products. We also contacted networking professionals who have used these products and asked them to share their experiences and their opinions. In selecting these four products, we tried to offer a representative sampling of some of the various approaches and philosophies that apply to LAN analysis.

- The Sniffer from Network General is one of the more talked-about protocol analyzers being used today. We asked Phil Fernandez, Manager of Computing and Network Services for the Stanford University Medical School, to describe how he uses the portable Sniffer to keep the Medical School's eight disparate Ethernet systems running.
- ARC-Monitor from Brightworks Development is a software solution to ARCNET troubleshooting. In this profile, Paul Cosgrove tells us how ARC-Monitor works and how he used this software package to quickly isolate a problem in his client's ARCNET networks.
- LANProbe from Eon Systems makes its debut this month. Unlike other protocol analyzers or less sophisticated troubleshooting devices, this hardware/software combination turns an IBM AT into a dedicated workstation that monitors the flow of data over Ethernet and helps highlight problems before they bring a network to its knees. LANProbe doesn't start shipping until some time this month, but we did get some input from Eon's beta test sites to determine how effective this approach is in keeping network traffic flowing.
- LAN Detector is another protocol analysis device for Ethernet from Micom InterLAN. In this profile, we discuss the inner workings of this hardware board/software system and we talk to George Ducharme, the network specialist for Versatec and a first-time user of protocol analysis.

These four products have striking similarities as well as marked differences. They represent only a sampling of the commercial solutions that are available for analyzing local area networks. Please use the reader service card or drop us a line to let us know if you find these kinds of product profiles valuable. □

Using The Sniffer
To Diagnose LAN Problems

When you have to support TCP/IP, XNS, and DECnet in the same environment, diagnostic tools are an absolute necessity.

by Phillip M. Fernandez

Local area networking systems are at the leading edge in developing multiuser, microcomputer, and supermicro applications. And while LANs enable entire workgroups to gain the advantages of distributed processing, shared file storage, and centralized routine operations, networks tend to become more complicated as they grow in size. The need for LAN diagnostic tools is emerging as a necessity for corporate network system developers, manufacturers of network-compatible hardware and software, and independent network integrators.

Protocol analyzers are some of the most powerful of these diagnostic tools for local area networks. By making traffic over the LAN visible, integrators can instantly locate problems that were previously invisible. The technology is providing system integrators with a method to ferret out the source of network problems by incorporating into their LAN environment "network X-ray" instruments that provide a way to graphically see what is going on inside a malfunctioning network.

At the Stanford University Medical School, we employ a protocol analyzer as part of the complex, multivendor network environment to help analyze, fine tune, and troubleshoot network functions. The Medical School network serves a user community of 3500 people in 15 buildings throughout the campus, or almost one million net square feet. Presently, approximately 400 users utilize eight Ethernet LANs linked in a TCP/IP internetwork. These networks support VAX hardware, Sun workstations, as well as PCs, and utilize TCP/IP, XNS, DECnet, and other network protocols. We have found the protocol analyzer an ideal tool to troubleshoot and maintain the network, which is rapidly growing as new users are added every week.

But a protocol analyzer is only one of several network diagnostic tools currently used at the Medical School. At the lowest level, a Tektronix Media Analyzer is used to test the basic electrical connectivity of the network. If complex low-level problems occur, a Tektronix Time-Domain Reflectometer (TDR) is used to identify, locate, and correct network problems, such as open and short circuits, faulty transceiver taps, and other problems.

Phillip M. Fernandez is Manager of Computing and Network Services at the Stanford University Medical School in Palo Alto, California.

When more complex or subtle problems occur with network communications, a protocol analyzer is used. For example, tests with the TDR may show that the network is physically sound, but hosts on the network may still have difficulty communicating. In this case, common problems include incorrect address assignments, faulty software configurations, and incompatible network protocols. The protocol analyzer can be used to capture and decode Ethernet packets crossing the network and aid in isolating these problems.

Sniffing Out LAN Problems

To meet the complex diagnostic needs of the Stanford Medical School, I decided the Sniffer, manufactured by Network General of Mountain View, California, would be the best tool to do the job. The Sniffer has the capability to support the Medical School's various network protocols, and provides excellent support at all levels of the OSI protocol model. The portability of the Laptop Sniffer version was a bonus in tracking problems across departments throughout 15 buildings.

The Laptop Sniffer is a self-contained unit housed in a Toshiba T3200 AT-compatible with a 12-MHz 80286 microprocessor and a 40-MB hard disk drive. It collects, records, and analyzes network data and monitors real-time network performance. Included with the Sniffer is a three-window, multilevel menu system that views one protocol level or all subsequent levels in summary, detailed, or hex view, as well as providing a byte-by-byte English interpretation of network protocols, synchronized scrolling of multiwindow data displays, and a high-speed capture mode that instantaneously captures and displays every frame.

In choosing a diagnostic system, it is important to consider support for the OSI protocol model which defines seven layers of network protocols (see Mark Miller's article, "Troubleshooting Local Area Networks Using the OSI Model," page 18). The lowest layers, layers one and two, provide physical connectivity and basic data transfer. Layers three through five provide for internetworking, reliable transport, and session management; layers six and seven provide applications service.

The Sniffer, for example, works at all of these seven layers. When working with the TCP/IP protocol, the Sniffer is able to decode layer three IP packets and layer four TCP and UDP packets. In addition, the Sniffer supports a number of layer seven application protocols such as Telnet, FTP, as well as Sun's NFS and Yellow Pages.

The Sniffer allows the appropriate protocol layer to be selected and displayed. If there are problems with internet routing, the Sniffer can display the IP portion of each captured network packet. In this mode, the Sniffer decodes and displays critical information, such as source and target IP address, in plain text. If there are problems with application protocols, such as Network File System (NFS, a TCP/IP-based distributed filing protocol), the Sniffer's display can be shifted to show just the relevant NFS application portion of each network message, such as file names and file handles. The user interface allows the protocol layer and level of detail to be tuned to the specific need.

Protocol Analysis In Practice

At the Stanford Medical School, we use the Sniffer about once a week,

Figure 1. The Sniffer summarizes network traffic in skyline profiles. The upper graph shows traffic level in frames/seconds; the lower, the number of active stations.

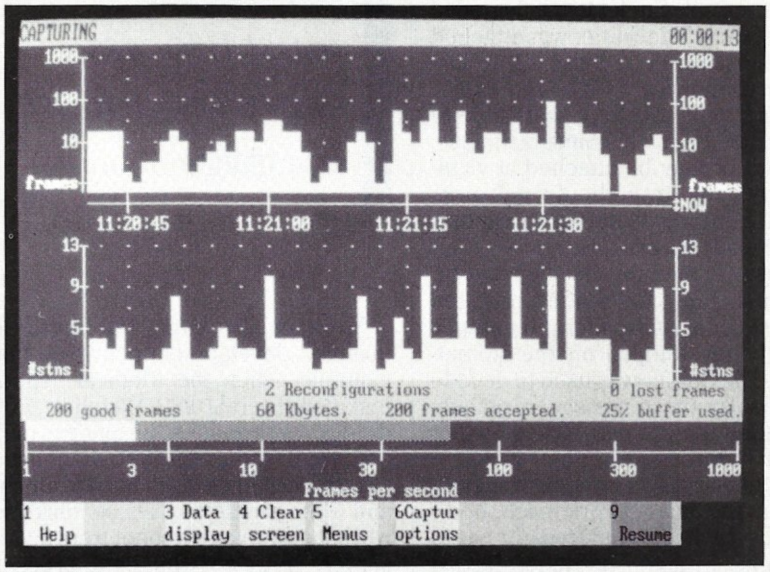
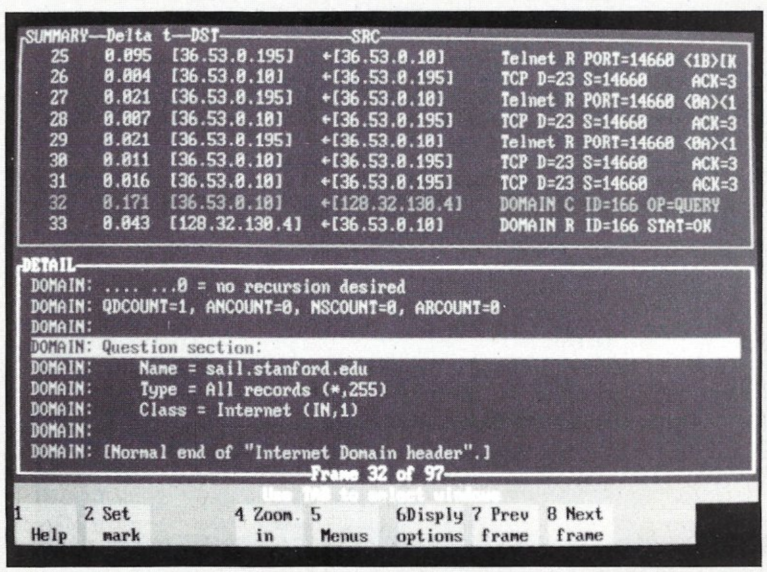


Figure 2. The Sniffer can look behind the gateway in an Ethernet environment by showing addresses embedded in higher level protocols.



mostly to respond to specific network problems and to optimize performance. To take one example, severe periodic performance degradation was reported on one subnet of the Medical School network. We on the network engineering staff suspected that the performance problems might be caused by unnecessary broadcast messages being transmitted on the subnet. All hosts on a subnetwork must process every broadcast message, so if there are excessive or spurious broadcast messages, all hosts will devote significant amounts of time to processing these messages, thus degrading their performance.

Working on this theory, we hooked up the portable Sniffer to the degraded subnetwork. The Sniffer was attached to the subnetwork by detaching a transceiver cable from an IBM PC on that network, and attaching that cable to the Sniffer. Using this same technique, the Sniffer may be attached at virtually any point throughout a network.

The Sniffer was set up to capture only broadcast packets, and was left to passively monitor the subnetwork. Within as hour, the Sniffer indicated that "storms" of broadcast messages were being dumped on the subnetwork. The Sniffer was used to decode and display these messages, quickly indicating the problem. It seems that 256 IP Address Resolution Protocol (ARP) messages were being transmitted on the subnetwork each hour, one for each host address in the 0 through 255 range. The Sniffer showed that these messages were being sent by an Inter-Network Router (INR) attached to the network. This INR executes software we had written for the Medical School, so we quickly focused our attention on this software. With the help of the protocol analyzer, the software problem causing the broadcast storm was isolated and corrected, with the result that performance was restored.

All hosts on a subnetwork must process every broadcast message . . .

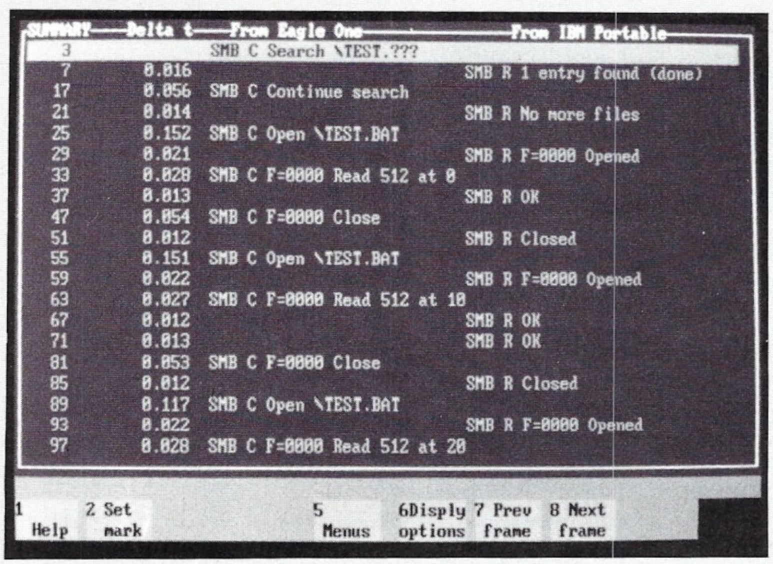
We have also found the Sniffer is useful in development activities. For example, when the Medical School installed several DEC VAXstation 2000 diskless workstations, the Sniffer's multiprotocol and multilayer diagnostic capabilities quickly resolved protocol and NFS problems. These diskless workstations were to boot from a VAX server; however, the boot process failed to work. Using a spare transceiver, we attached the Sniffer to the subnetwork with the VAXsations. It was then set to capture packets sent between one VAXstation and the server, and the boot process was repeated. The Sniffer recorded, decoded, and displayed the network interchange between the VAXstation and the server. We found that low-level DECnet Maintenance Operations Protocol (MOP) messages were being sent by the VAXstation, but were not being processed by the server. With this information, DEC software service was contacted and we learned that a kernel configuration option had to be set within the server.

With this problem corrected, the boot process was again attempted, but it failed a second time, this time much later in the boot process. The Sniffer was again used to track the network interchange between the VAXstation and the server. The problem was found to be an NFS mismatch between the VAXstation and the server. The Sniffer decoded the NFS application-layer packets, which pointed to a problem with file access permission on the server. When the permissions were set properly, the VAXstation booted correctly and was fully operational.

The Sniffer has also helped diagnose incompatibilities during the installation of Metaphor workstations. There were some XNS protocol problems between Metaphor workstations and file servers. In these examples, the Sniffer was used to capture essential data in the network, which was then provided to the responsible vendor who made the corrections.

In a highly complex, multivendor network environment like the one we maintain at Stanford, diagnostic tools have become an absolute necessity. Protocol analysis tools like the Sniffer provide a way to dramatically extend our diagnostic capabilities, optimizing the functionality and performance of the network. □

Figure 3. The two-station presentation available in The Sniffer makes it easy to track command/response situations (note 'C' and 'R' in the display), in this case DOS-level SMB frames.



Did you find this article particularly useful? Circle number 5 on the reader service card.

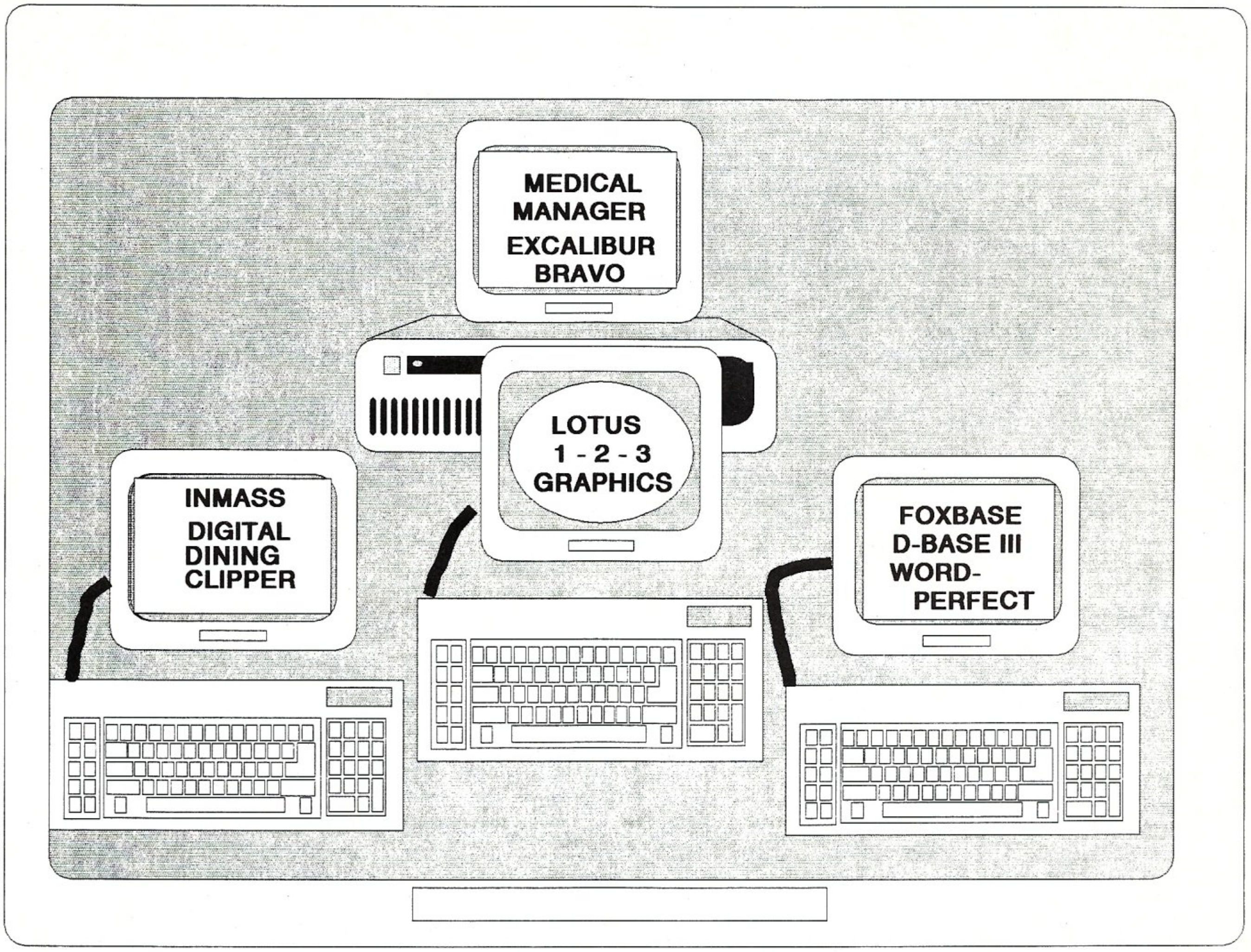
Product Information

The Laptop Sniffer $15,000

Network General
1945A Charleston Road
Mountain View, CA 94043
(415) 965-1800
Circle reader service #252.

THINK SMART, NOT DUMB

For your next multiuser system



SMARTSCREEN™

Smartscreen, an enhanced version of Concurrent Dos 386, and Viewport Technology's new monochrome multiplexer board, supports multiple Hercules-compatible monochrome monitors and regular keyboards, instead of dumb terminals, for your multi-user workstations.

* Cost-efficient TTL monitors and keyboards can be used
* Each station has its own video RAM
* Direct video link for highest speed possible
* No emulation for greater PC-DOS compatibility and speed

For more information call Viewport Technology, Inc. at (408)748-1495 and Concurrent Controls Inc. at (415) 873-6240. *If you mention this ad, you'll receive a 10% discount on your first Smartscreen order.* Dealer, distributor and OEM inquiries welcomed.

CONCURRENT CONTROLS, INC.

A Software Engineering Company
925A Linden Avenue, South San Francisco, CA 94080
Phone (415)873-6240, FAX (415)873-6091

CIRCLE 53 ON READER SERVICE CARD.

ARC-Monitor Plus: Diagnostic Software for ARCNET Systems

What do you do when your ARCNET LAN hangs up? Boot this troubleshooting package and let the software isolate the problem.

by Paul Cosgrove

Have you ever spent hours trying to isolate a problem within an ARCNET network? When the network goes wrong, the phone starts ringing off the hook with complaints from network users: "Why is the system so slow?" "Is the system down? I can't get in." You have to find the problem and find it fast. But where do you start? With the file server, the cables, the hubs, or with the workstations themselves?

This was just the situation I faced when I received a call from one of my Denver clients. This particular government office maintains an ARCNET network of 35 workstations, six of which are closely configured in one laboratory. When the network would slow to a crawl, the resident engineer would start switching active hubs, which seemed to solve the problem for a time but the system would slow to a crawl again within a few days. Were the active hubs the source of the problem, or was the source of the trouble somewhere else?

The question was where to begin? Luckily, I had help in the form of a software tool called the ARC-Monitor Plus, which can track activity on any ARCNET network. With the aid of ARC-Monitor Plus and some legwork, I was able to isolate the source of the network slowdown. It turned out the active hubs were fine, it was one of the ARCNET boards that was bad. The act of changing the hub would disconnect the faulty workstation and that was what improved the network's performance. Once the faulty network card was isolated and replaced, the system was back on line and running smoothly.

About ARC-Monitor Plus

The ARC-Monitor Plus software runs from any workstation to provide both active and passive diagnostics on ARCNET LANs to isolate the source of hardware faults and traffic over-

Paul Cosgrove is owner and operator of Dr. P.C. Housecall, a network consulting firm based in Northglenn, Colorado.

Figure 1. The ARC-Monitor Plus opening menu.

```
        Westcon Associates Inc.
     Eastchester, New York 10707
         (914) - 779 - 4773

        Version Number -1.4
        Serial Number -  02313

======== ARCMONITOR - Plus ========

      Node and Packet Display
        Token Timing Graph
      Activity Level Display
          Test this Card
               Exit
               Help

====== the node address is  37 =====
```

Figure 2. The Node and Packet Display gives the status of each workstation.

```
                    ARCMONITOR - Plus
                 Westcon Associates, Inc

==============<node map>=================
node   user     packets | node   user     packets     page totals (1-8)

 0 WORKSTATIO      0    | 16 WORKSTATIO      0        1    0003649
 1 WORKSTATIO   2722    | 17 WORKSTATIO      0        2    0002720
 2 WORKSTATIO   3650    | 18 WORKSTATIO      0        3    0000000
 3 WORKSTATIO      0    | 19 WORKSTATIO   1307        4    0000000
 4 WORKSTATIO      0    | 20 WORKSTATIO      2
 5 WORKSTATIO      0    | 21 WORKSTATIO      0        5    0000000
 6 WORKSTATIO      0    | 22 WORKSTATIO      0        6    0000000
 7 WORKSTATIO      0    | 23 WORKSTATIO      1        7    0000000
 8 WORKSTATIO      0    | 24 WORKSTATIO      0        8    0000000
 9 WORKSTATIO      0    | 25 WORKSTATIO    692
10 WORKSTATIO      0    | 26 WORKSTATIO      0     segment total  0006373
11 WORKSTATIO      0    | 27 WORKSTATIO      0     system  total  0006375
12 WORKSTATIO      0    | 28 WORKSTATIO      2     last reconf 10: 41: 16
13 WORKSTATIO      0    | 29 WORKSTATIO      0     total reconfigs  00001
14 WORKSTATIO      0    | 30 WORKSTATIO      0
15 WORKSTATIO      0    | 31 WORKSTATIO      0      Starting node :  0
```

loads. On the ARC-Monitor diskette there are three programs: ARCPLUS, USERMAP, and RECALARM.

The ARCPLUS program is the real tool you will want to use. ARCPLUS will help determine: 1) the capacity of your ARCNET network by displaying you how much data is currently being transmitted compared to network capacity; and 2) which workstations are causing network difficulties. It gives you a graphic representation of network traffic and helps you isolate the workstations that are sending faulty packets.

ARCPLUS lists each workstation by its address, along with the total number of packets sent from that workstation and the number of faulty packets sent. The program will display this information either in columns for direct comparison or graphically in a bar chart format. In fact, the program can be used to monitor network use to determine heavy traffic periods.

USERMAP is a program that helps set up groups and workstation names for use in the NODEMAP.DAT file of ARCPLUS. This program has no parameters and only returns the physical ID of nodes attached to the current server you are using, along with login names and station numbers assigned to the server.

RECALARM provides a network monitoring alarm. This memory-resident program is designed to stay in memory and alert the operator that a reconfiguration on the network has just occurred. There are four options that can be prompted from the first line of the page 0 screen: \b, \c, \e, and \t. Option \b will alert the users with a banner and a beep. Option \c is for CGA and EGA monitors and will change the color of the banner from white on black to red on black. Option \e will alert the user every time there is a reconfiguration (The default setting will trigger the alarm only if there are 10 or more reconfigurations within a minute.) Activating this option is useful when you want to make sure that a certain terminal is properly connected to the LAN. Option \t will extend the length of time that the banner will appear.

RECALARM is invoked before a normal network login. It can be loaded with AUTOEXEC.BAT before network calls are made. Normal DOS syntax is:

```
A:\>RECALARM\B\C\E\T
```

If the workstation user is currently running an application in the foreground and that software application does not use DOS function calls, RE-

Figure 3. Instructions to display the next group node.

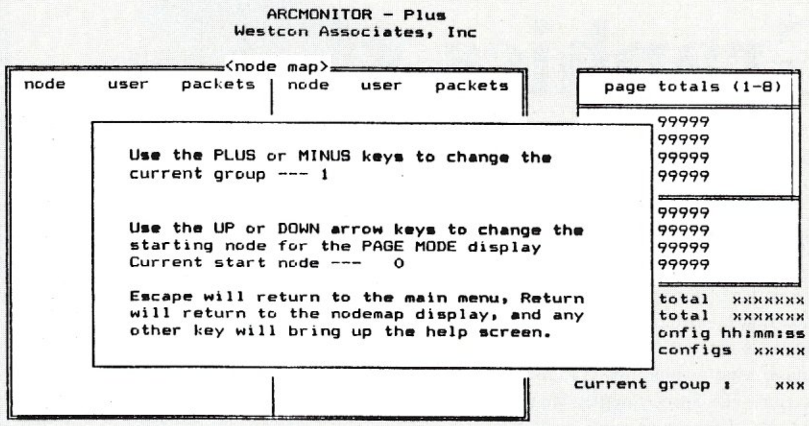


Figure 4. The display prompt for the "Test This Card" mode.

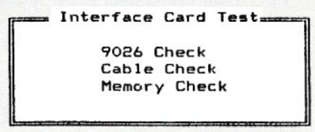


Figure 5. Address results for the "Test Card Mode."

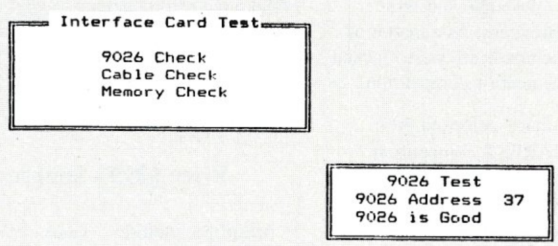
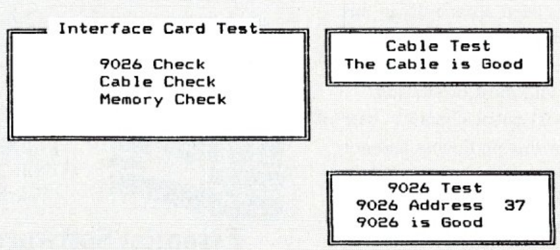


Figure 6. Cable test results for the "Test Card Mode."



CALARM will display the banner, if required, after the package session is over and DOS is invoked.

ARCPLUS in Action

Because of the nature of its operation, ARCPLUS must be run outside of the network shell at all times. Previous versions allowed you to run ARCPLUS under the shell, but doing so would bring down the file server. ARC-Monitor Plus Version 1.3 and later will not allow the user to load ARCPLUS under the shell, even accidentally. Instead, a message will appear that informs you that the shell is already loaded and the program will wait for a key to be pressed before causing a complete reboot. At the normal termination of an ARCPLUS session, a long boot will be invoked after a final keystroke is made.

> *Were the active hubs the source of the problem, or did the problem lie elsewhere?*

ARCPLUS is invoked on the workstation before the network shell is loaded. When invoked, a menu appears giving four different operating mode options: "Node and Packet Display," "Token Timing Graph," "Activity Level Display," and "Test This Card" (Figure 1). The main menu also displays the workstation address for the ARCNET card in the node.

Selecting the "Node and Packet Display," displays the status of the workstation on the network with a screen like the one shown in Figure 2. The screen can be changed to display the next group node by pressing the "+" or "–" keys on the keypad and following the instructions (Figure 3). Displayed on the screen are 32 node address numbers, the user name (which can be customized), and the number of packets transferred to and from that node. In Figure 2, nodes 1 and 2 have a high packet count because those are the ARCNET cards in the server. On the right hand side of the screen is the packet count for each page of 32 nodes. At the bottom right is displayed the total number of reconfigurations since the ARCPLUS program was started. This is the number to watch. Whenever a workstation creates its shell one or two reconfigurations normally occur, but if this number continues to increment there is a workstation with a bad ARCNET card.

The next step is to find which node is the culprit. Look for a node with a high number of packet transmissions. The large number of packets could be part of the normal traffic flow, or it could be the source of the problem. If none of the workstations are generating packets and the problem seems to be that the network can't establish handshaking, you can start turning off workstations one by one until the reconfigurations stop. In a large office this is easier to do with two people and an intercom.

Once you have isolated the workstation, the next mode to apply is "Test This Card." Selecting this mode displays a screen like that shown in Figure 4. This mode will test the ARCNET card, the cable connection, and the card's memory (See Figures 5, 6, and 6). This will isolate the problem and should tell you whether to replace the cable connector or the card itself. I run these card tests on the workstations at least once a year as part of my preventive maintenance routine.

This kind of software makes it a simple matter to isolate and diagnose a network problem by working from the workstations back to the server. With a little practice and harmless experimentation, it takes very little time and effort to fix the network when that phone starts ringing with complaints. □

Did you find this article particularly useful? Circle number 6 on the reader service card.

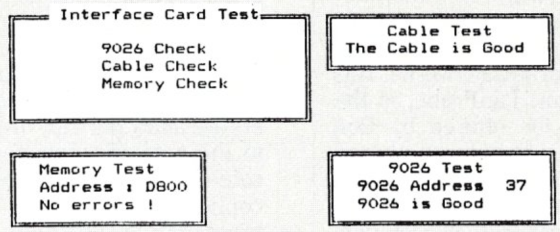
Product Information

ARC-Monitor Plus $295
E-Monitor (Ethernet) $125

Brightworks Development
P.O. Box 8728
Red Bank, NJ 07701
(201) 530-0622

Circle reader serivce #253.

Figure 7. Memory test results for the "Test Card Mode."



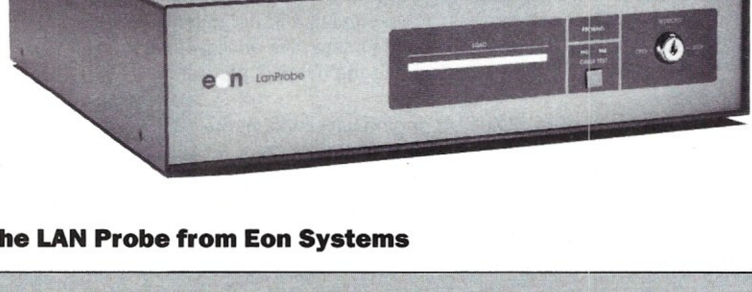
LanProbe: An Information System for Ethernet

This month, a new LAN monitoring product becomes available to help LAN managers design, install, and maintain Ethernet local area networks. This new LAN monitor, LanProbe, is the first product to be offered by Eon Systems, Inc., a new company based in Cupertino, California.

The LanProbe system is a hardware/software combination that uses dual NEC V50 processors (8 MHz) and the Intel 82586 LAN coprocessor. The system requires an IBM AT with EGA or VGA to act as a host computer, although the system is compatible with 386 and PS/2 hardware. It also requires 640K of memory, 450K of which is needed for the operating software, and a hard disk with 2 MB free after loading Microsoft Windows, which is the software platform for LanProbe. It requires DOS 3.0 or higher, Version 2.0 or higher of Windows, and a Windows-compatible mouse. The system also has an internal, 2,400-baud modem for dial-in data gathering to facilitate long-distance LAN diagnosis.

The LanProbe system itself is made up of three different parts:

- **LanProbe ME** is the hardware platform of the segment monitor. It attaches to the end of an Ethernet segment and monitors all the packets on that segment. It can be connected using thin or thick coaxial cable, twisted-pair cable, or fiber-optic cable. Once the packets are analyzed they are uploaded to the network management console via RS-232, Ethernet datalink connection, or modem.
- **Surveyor** is the other hardware option attached to the opposite end of the Ethernet system. The Surveyor's built-in Time Domain Reflectometer (TDR) makes it possible to accurately map node and device positions and provides a physical representation of the network. It also makes it possible to pinpoint cable breaks and trouble spots on the map right down to the number of feet from the nearest node.
- **ProbeView 1.0** is the interpretive software that presents network node and traffic information to the LAN administrator. Based on Microsoft Windows, it maps node information for the network, and charts and graphs statistical data describing network performance. An embedded database is used to organize information by node, segment, or network. This data can then be copied and pasted into other applications running under Microsoft Windows.



The LAN Probe from Eon Systems

There are six critical areas that LanProbe monitors and graphs: the percentage of system utilization; number of packets sent per second; number of bytes per second; number of broadcasts per second; number of errors per second; and the number of collisions per second. Thresholds of performance are set for each parameter, and an alarm will be triggered if the threshold is exceeded. These levels of performance also are tracked in RAM and stored in an event log on the hard disk. Once captured, the data from the log can be automatically graphed by the ProbeView software to highlight problems.

Dr. Joseph Adler, president and CEO of Eon, emphasizes that the LanProbe is not a protocol analyzer or conventional LAN troubleshooting device. Unlike protocol analyzers that require substantial network expertise to interpret, LanProbe was developed for LAN managers as a smart system with the analytical intelligence provided in the software. The graphic presentation format makes it relatively easy to interpret LAN performance data.

Unlike troubleshooting systems, LanProbe is a dedicated system that remains online to continually monitor the Ethernet LAN. This not only makes it possible to isolate a fault when a failure occurs, but it makes it possible to graphically see where performance problems may arise at some point in the future.

"This system is dedicated, just as a dashboard on a car is dedicated," Adler explains. "Can you imagine driving down the highway and having to plug in a speedometer periodically to see how fast you are going? It is critical to have a real-time, ongoing method of tracking network performance."

For example, at one of the six beta sites used to test LanProbe, continual monitoring over a period of time showed that the high level of traffic over the Ethernet backbone did not fall off in the evenings or on weekends. LanProbe graphically illustrated that transmissions of 400 packets per second continued through Saturday night, indicating that "idle" background traffic emanating from the 150 nodes on the network was congesting the system. The LAN manager in charge plans to solve this problem by replacing the simple repeaters used to interconnect the network with MAC-layer bridges, thus eliminating background traffic.

Another beta site reported that its network was "sluggish." The source of the slowdown was excessive broadcast packets that took up from 10 to 40 percent of all network traffic. Whenever a broadcast packet is sent, every node has to decode the protocol into software in case the broadcast packet was a control message. If the network hardware does not filter multicast packets, those packets are treated as broadcasts that must be read and decoded; therefore every multicast packet or broadcast consumes processing power. Using the LanProbe, this broadcast problem was graphically highlighted, displaying both the number of broadcasts and the time of day the broadcasts were at their peak. Armed with this information and historical data about trends in network traffic, the source of the traffic was determined to be a new database package being used by Accounts Payable.

The LanProbe ME and ProbeView 1.0 software will become available this month. The Surveyor with built-in TDR is expected to follow within the next few months. Sources at Eon Systems indicated that similar LanProbe products will be developed for ARCNET, Token Ring, and other LAN configurations in the future. □

Product Information

LanProbe	$8,000
ProbeView	$5,000
Surveyor	TBA

Eon Systems, Inc.
10601 S. De Anza Blvd., Suite 305
Cupertino, CA 95014
(408) 252-5440
Circle reader service #258

Figure 1. ProbeView Software provides a "snapshot" of network utilization.

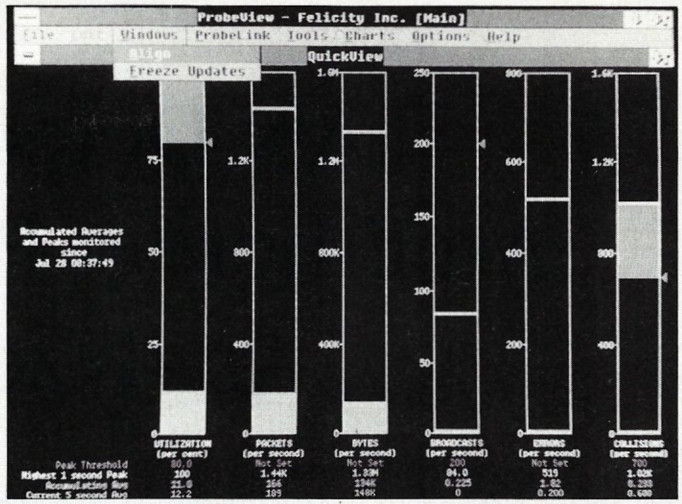


Figure 2. Using Windows, LanProbe simultaneously displays a variety of data.

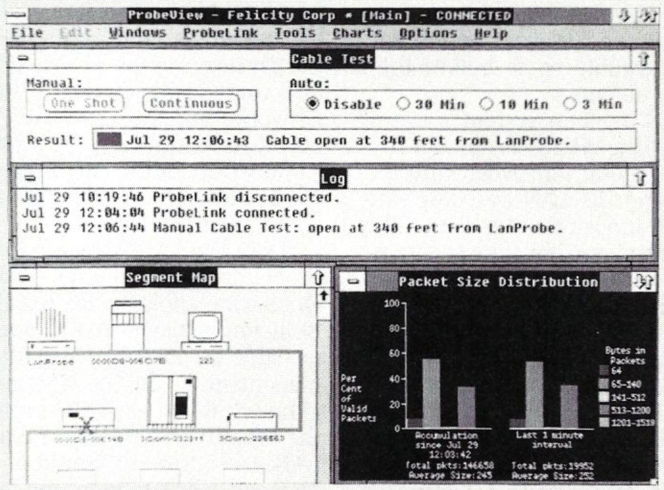
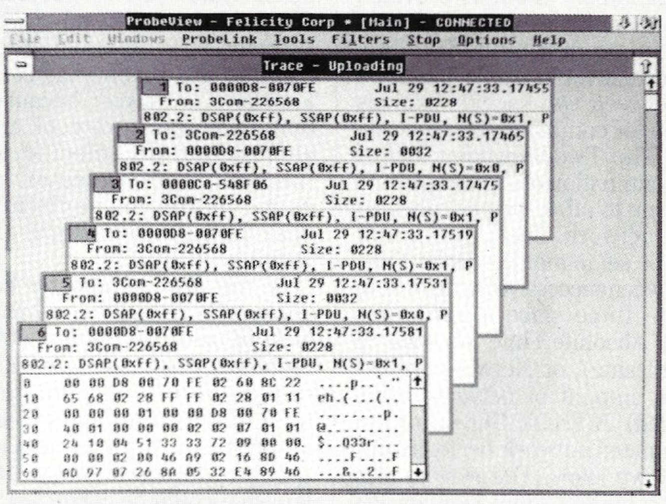


Figure 3. Packet tracing is included as part of protocol analysis.



LAN Detector: An Ethernet Protocol Analyzer

Many network managers are seeking an inexpensive protocol analysis solution for Ethernet. LAN Detector from MICOM-Interlan is a hardware/software combination that can be inexpensively installed in an IBM AT to monitor Ethernet traffic and highlight trouble spots before they become problems.

The LAN Detector system is a combination of the NP600 Protocol Processor board that interfaces with the network and the LAN Detector software that interprets the data. The system operates in real time and translates embedded protocol information into English. LAN Detector is compatible with a variety of network protocols, including Novell, XNS, TCP/IP, Sun PC-NFS, and DECnet.

The NP600 captures every frame of network traffic, and the software filters the data by address, protocol, bit pattern, frame error condition, or any combination. Specific capture and display configurations can be programmed for ease of analysis, and a Display Summary mode is provided to present multiple options to help diagnose network failures. The Two-Station format, for example, monitors traffic between two specific stations on screen for command/response diagnosis. The Two-Viewpoint format allows examination of each frame and comparison to other frames captured from the network. The LAN Detector can also be set to monitor Delta Time (time between successive frames), Relative Time (time since a frame was marked), Absolute Time (time stamp for each frame), or Network Utilization (the amount of network bandwidth used). A Traffic Generator feature simulates network performance under heavy loads. The system also features a built-in Time Domain Reflectometer (TDR), has a variety of output options, and has modem capability for remote diagnostics.

George Ducharme, the resident Network System Specialist for Versatec, a manufacturer of electrostatic plotters based in Santa Clara, California, has been using the LAN Detector for the past six months. Versatec has two interconnected, fiber-optic Ethernet networks running XNS and TCP/IP protocols to link more than 200 workstations, including 30 diskless Sun workstations, and six CAD servers among four buildings.

"I use the LAN Detector to isolate problems and keep track of activity," Ducharme explains. He has the LAN Detector installed in a Compaq 286 portable computer as a virtually permanent node on the XNS network, although it can easily be switched to the TCP/IP LAN. Mostly, he uses the LAN Detector to analyze traffic between workstations and between servers.

Although he never used a protocol analyzer before, Ducharme says he doesn't know "how I got along without one." Ducharme admits that LAN Detector may not have all the best features, and that he had considered Excelan's LANalyzer because it is a more powerful protocol analyzing alternative. He selected MICOM-Interlan's LAN Detector because its menus and on-line "Help" facility made it easy to use without extensive training.

In the few months he has been using LAN Detector, he has already isolated and corrected a number of problems. For example, remote workstations were having difficulty trying to dial in to a gateway because of identification/authentication errors. LAN Detector quickly isolated the problem so it could be corrected.

Ducharme thinks the greatest strength of protocol analysis is in preventive maintenance. "Generally, problems start creeping into a LAN," he explains. "Errors start slowing down the network —authentication problems, error messages —until it snowballs into a complete collapse. Monitoring reveals those problems before you hit the collapse. You can't start troubleshooting your network once it goes bad. You have to look at it while it is still running."

To illustrate his point, Ducharme notes that on Versatec's TCP/IP network, diskless node traffic was reaching 40 to 50 percent. "When I observe that kind of traffic, I see that I have to start putting on subnets to get the diskless nodes off the backbone," he said.

Ducharme did look at other LAN management approaches before deciding to make the expensive investment in a protocol analyzer. With the additional purchase of the Compaq 286 computer, Ducharme's total cost exceeded $20,000, but he feels the investment was worth it. "The expense was hard to justify, but consider the cost of having to take 30 engineers off-line for six hours. The downtime that can be saved with a protocol analyzer makes it definitely worth it." □

Product Information

LAN Detector $10,995

MICOM-Interlan Division
155 Swanson Road
Boxborough, MA 01719
(508) 263-9929
(800) 526-8255

Circle reader service #262

Figure 1. A skyline display summarizes real-time network performance.

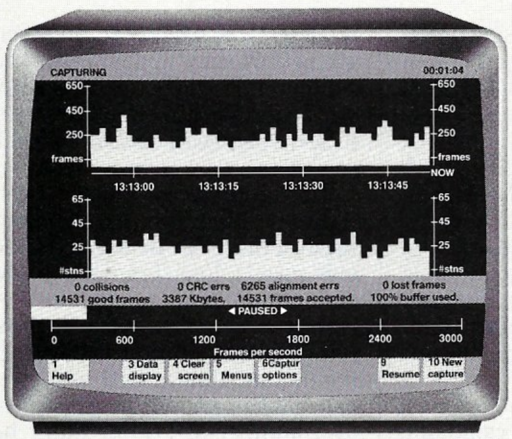
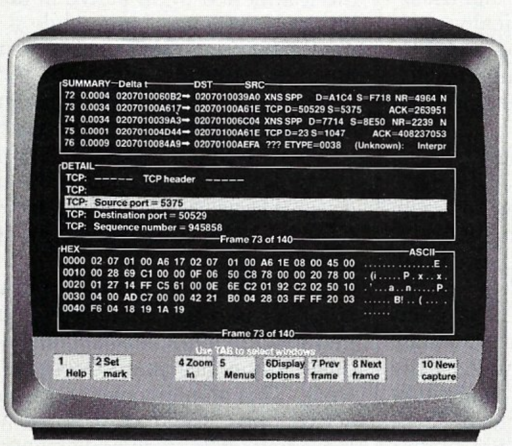


Figure 2. A protocol interpreter gives in-depth information.



Comparing ARCNET Cards

A comparison of hardware approaches to creating an ARCNET LAN topology.

by William Wong

ARCNET is a token-based network originally developed by Datapoint for its ARC computer system. The system was quite advanced at the time, but PCs quickly became the workstation of choice with IBM's support. The ARC computers were not PC-compatible. However, both Datapoint and ARCNET have survived.

ARCNET became one of the first popular local area network links. ARCNET support chips were developed by Standard Microsystem Corporation (SMC). SMC supplies these chips to a number of ARCNET board manufacturers, and manufactures its own line of ARCNET boards as well. ARCNET was originally developed using coaxial cable. Fiber-optic links were also available, but until recently twisted-pair wiring was not an option. ARCNET cards are supported by a number of LAN operating systems, including Novell's Advanced NetWare and Banyan VINES.

ARCNET has a number of advantages, including ease of installation and maintainability. It is also a standard supported by a large number of vendors.

An ARCNET Overview

ARCNET was originally designed as a modular LAN link using hubs and coaxial cable to connect up to 255 computers together. Data is sent at a rate of 2.5 megabits per second (mbps) in packets of up to 512 bytes using a modified token-passing protocol similar to that used by the IBM Token Ring. Although the data packet is the same size as a standard PC disk sector, the match is not as good as it seems since ARCNET imposes a 4-byte overhead, causing the largest data packet to be 508 bytes. However, the packet size can be varied for more efficient network utilization. ARCNET is also self-reconfiguring so computers can be attached and detached, or be powered on or off from the network at any time.

ARCNET was a major technical achievement when it first came out. It might appear to be a bit dated in this age of 10-mbps Ethernet and 100-mbps fiber-optic networks, but ARCNET is still viable and, in many cases, preferable for today's LAN. LANs may have thousands of PCs, but a closer look will often show that these PCs are logically or physically connected in work groups of under a hundred PCs. Large systems usually have the servers interconnected on a common LAN.

ARCNET can be used in large LANs with more than 255 PCs if the network servers act as bridges between workgroups. ARCNET also fits very well into small- and medium-sized LANs with as few as four PCs.

ARCNET's throughput is greater than StarLAN's 1 mbps but less than IBM's 4 mbps Token Ring or Ethernet's 10 mbps. But raw bandwidth is only part of the story. Ethernet and StarLAN both use a collision detection protocol as opposed to the token-passing schemes employed by ARCNET and Token Ring. The token-passing protocols are more efficient, and performance under heavily loaded conditions is also better, which is why the Token Ring is often compared to Ethernet and why ARCNET performs so well in a LAN environment.

The layout of an ARCNET system is an unrooted tree that is noncyclic (that is, there are no loops, as shown in Figure 1). Each branch or segment of the tree is connected to a hub. Hubs can be either active or passive. Active hubs are powered devices that regenerate timing and amplify signals allowing a hub to be connected to a number of PCs or other hubs. Active hubs normally have 4 to 16 connections. Passive hubs are not powered and do not regenerate or amplify signals. Passive hubs are limited to 4 connections. Connections are made using RG-62/U coaxial cable. A segment attached to an active node can be up to 2000 feet long, while a segment attached to a passive node must be under 100 feet in length. Active nodes can be attached to active hubs but passive hubs can only be attached to active hubs or ARCNET adapters. Unused connections on an active node do not have to be terminated, but passive node connections must be terminated.

The minimum distance between adapters is 6 feet and all PCs must be within 20,000 feet (4 miles) of each other if they are all on the same network. This would require at least 10 active hubs. Even so, it does allow large networks to be constructed using only ARCNET adapters and hubs.

The original version of ARCNET, just described, is supported by al-

William Wong is president of Logic Fusion, Inc., a systems software development firm located in Morrisville, Pennsylvania.

most all vendors of ARCNET adapters. Two newer implementations are based on fiber optics and twisted-pair cables. All versions operate using the same protocol running at 2.5 mbps.

Up to 10 PCs can be attached to either a coaxial or twisted-pair daisy-chain segment. All adapters connected to a daisy-chain segment must use high-impedance drivers. The high-impedance coaxial adapters are attached using BNC T-connectors such as those used with thin-wire Ethernet. Twisted-pair adapters have two connectors, with cable being daisy-chained through the adapter card. The connectors are standard RJ-11 modular telephone jacks. It is possible to use normal telephone extension cords for short runs, but twisted-pair cable must be used for long runs. Each end of a twisted-pair chain must be terminated. The terminators are resistors attached to modular connectors that plug into unused connectors on the adapter boards.

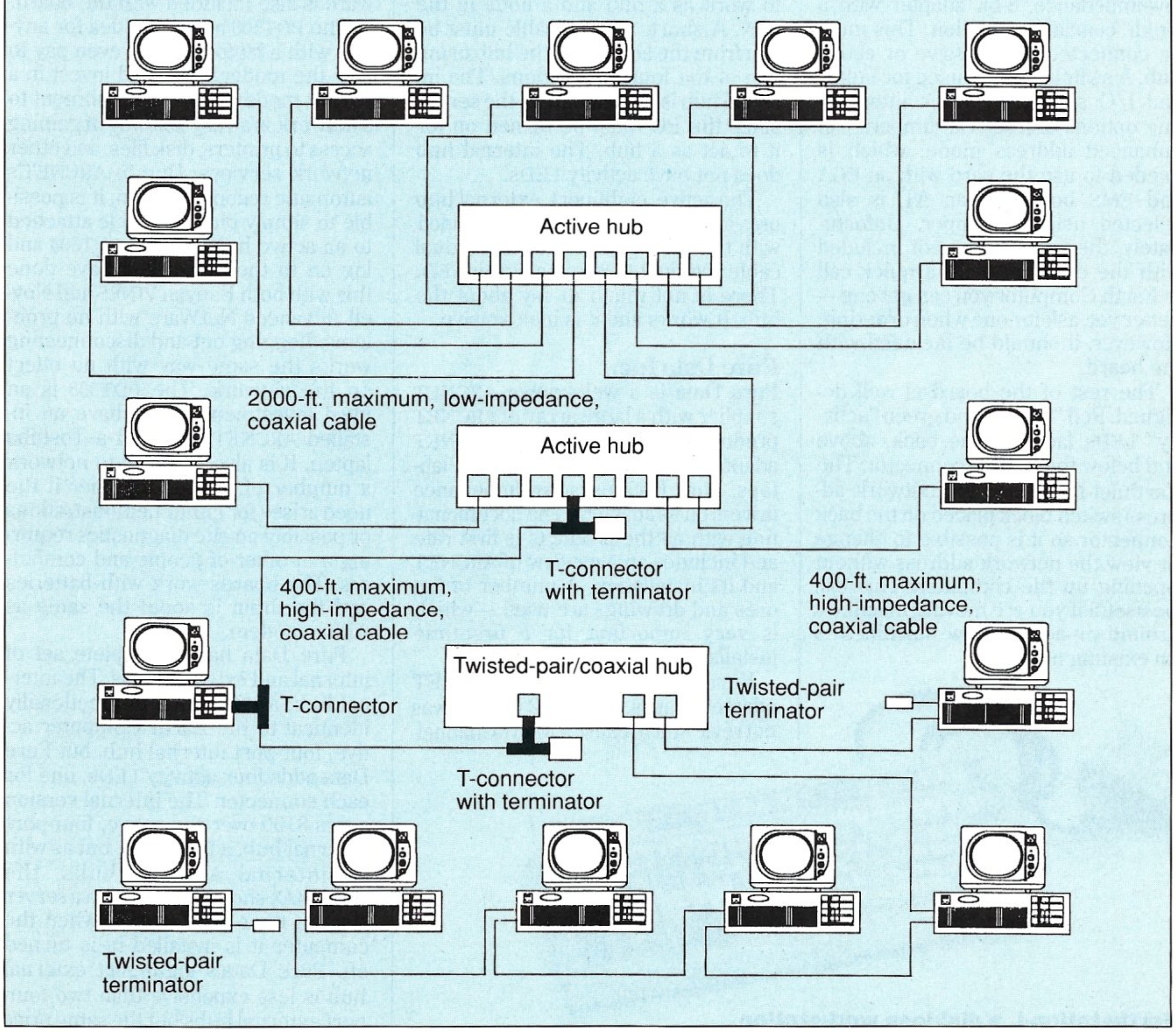
Each type of ARCNET adapter uses the same protocol at the same speed (2.5 mbps), regardless of the type of cable used. This allows an ARCNET LAN to include various types of adapters and cables. The conversion between different cable types is done at a special hub, which normally supports two types of adapters. The SMC two-port, twisted-pair, high-impedance coaxial hub is an example.

The consistency of this interface is easy to understand after looking at the various ARCNET adapter boards. They all tend to be based on the same SMC or NCR chip set, and each board has a noticeable hybrid transceiver circuit. Although the transceivers look the same, there are different ones for each adapter type.

The consistency is repeated in the hardware interface as well. Each adapter has a 2-kilobyte, dual-port memory buffer and an optional 16K boot ROM (except for the Toshiba adapter from Pure Data). Interrupts and I/O ports are selectable. This makes all the boards almost completely plug-compatible with each other. The differences occur because of extensions made by various vendors.

One area of concern with ARCNET cards is the placement of the 2K buffer. The default memory segment for most network software supporting ARCNET cards is at D000 hex. This is not a problem unless you have an AT with an EMS and an EGA board that also like to sit at D000. Unfortunately, many ARCNET adapters can only adjust the top 4 bits of the memory address, so you can put the buffer at B000, C000, D000, and so on. This leads to two alternatives: Don't use the ARCNET, EGA, or EMS board, or use only 512K of main memory and put the adapter

Figure 1. A typical ARCNET local area network



at A000. The latter occurs because D000 is the only 64K block free on an AT with a full 640K. Luckily, most of the newer ARCNET boards address this problem and let you place the buffer almost anywhere. An AT with an EGA and an EMS board at D000 still have a number of buffer slots at C400, C800, and CC00.

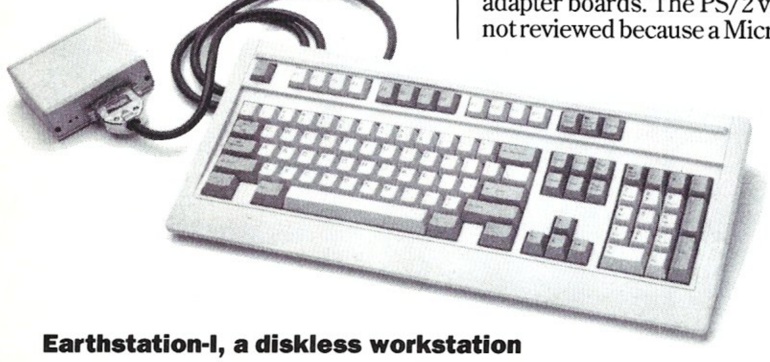
The ability to mix adapter types and adapters from different vendors in the same LAN can be a great advantage, especially in a LAN that is growing or changing. The appropriate adapter or cabling can be used as necessary.

Earth Computer Technologies

Earth Computer Technologies has some of the lowest priced ARCNET cards and hubs available. They also sell a keyboard that contains a complete PC, including video adapter and ARCNET interface (see the photo). All you add is the LAN cable and a monitor.

The Earthnet-I ARCNET board is a low-impedance, 8-bit adapter with a single coaxial connection. This must be connected to a passive or active hub. A switch block is used for buffer and I/O selections, while interrupt line options are set via jumpers. An enhanced address mode, which is needed to use the card with an EGA and EMS board on an AT, is also selected using a jumper. Unfortunately, the jumper was not included with the card, but with a quick call to Earth Computer you can get one — better yet, ask for one when ordering. However, it should be included with the board.

The rest of the board is well designed. Red "status" and green "activity" LEDs face out the back, above and below the coaxial connector. The Earthnet-I also has the network address switch block placed on the back connector so it is possible to change or view the network address without opening up the computer. This can be useful if you are moving machines around or adding new machines to an existing network.



Earthstation-I, a diskless workstation

Earthnet-I is very inexpensive — only $199, and that is list price. The documentation is a minimal 13 pages, and part of that is a disclaimer and FCC warning. However, it completely covers the operation of the board itself, including a description of both the AT/EGA/EMS, problem and solution. The board is capable of placing the RAM buffer at C400 and C800. The optional boot ROM can also be placed at CC00. The main thing missing from the documentation is a good overview of cabling and interconnection details such as the maximum segment length. A first-time ARCNET installer may run into problems if there is no other source of information.

The active, four-port, internal PC hub is a short card with four coaxial connectors. It plugs into an 8-bit slot on any computer (even one that is not attached to the network). The board draws power from the PC. The board does not contain an ARCNET adapter, so you will need another slot for the Earthnet-I board if the PC is to work as a hub and a node in the LAN. A short, six-foot cable must be run from the adapter to the hub using one of the four connections. The internal hub is often found on the server since the PC must be turned on for it to act as a hub. The internal hub does not have activity LEDs.

The active, eight-port, external hub uses an external power supply included with the hub. The hub has individual cable activity LEDs and a power LED. There is not much to say about the hub. It works and it is inexpensive.

Pure Data Inc.

Pure Data is a well-known ARCNET supplier with a large array of ARCNET products, including a unique ARCNET adapter for the Toshiba portable laptops. Pure Data uses low-impedance drivers on its adapters. The documentation with all the products is first rate and includes an overview of ARCNET and its installation. A number of figures and drawings are used —which is very important for a first-time installation.

Pure Data has three ARCNET adapter boards. The PS/2 version was not reviewed because a MicroChannel-based PS/2 was unavailable for the network. The other two boards include the PDI508, which is a short card, 8-bit adapter for the PC/XT/AT and the PDT508 for the Toshiba T1100, T1200, T3100, and T5100 laptop computers. The PDI508 has both a board and cable activity LED on the back connector along with the network address switches. The documentation addresses the AT/EGA/EMS problem, and the memory buffer can be set up like the Earthnet-I. An empty socket is included for a boot ROM.

The PDT508 plugs into the modem port of a Toshiba laptop. This means you lose the internal modem except on the T1200 with the factory-installed internal modem. The board installs easily and has only the coaxial connector and a power switch on it. The network address is set up via software that is supplied. The memory and I/O address are fixed, which is not a problem on the laptops since nothing else can be installed in the same place anyway. Diagnostic software is also included with the board.

The PDT508 is a great idea for anyone with a laptop. It may even pay to lose the modem port and invest in a pocket modem. Direct attachment to a network is a very fast way of gaining access to printers, disk files, and other network services. Due to ARCNET's automatic reconfiguration, it is possible to simply plug the cable attached to an active hub into the PDT508 and log on to the network. I have done this with both Banyan VINES and Novell Advanced NetWare with no problems. Logging out and disconnecting works the same way with no effect on the network. The PDT508 is an ideal investment if you have an installed ARCNET LAN and a Toshiba laptop. It is also practical to network a number of laptops together if the need arises for either demonstrations or possibly on-site diagnostics requiring a number of people and computers. The boards work with batteries and the drain is about the same as with a modem.

Pure Data has a complete set of internal and external hubs. The internal hub, a PDC504AX, is functionally identical to the Earth Computer active, four-port internal hub, but Pure Data adds four activity LEDs, one for each connector. The internal version saves $100 over the active, four-port external hub, a PDC504AI, but as with all internal ARCNET hubs, the PDC504AX should be placed in a server since it does not function when the computer it is installed in is turned off. Pure Data's eight-port external hub is less expensive than two four-port external hubs but the same price as two four-port internal hubs. Both the four- and eight-port external hubs have individual activity LEDs.

Standard Microsystems Corporation

Standard Microsystems Corporation (SMC) is well known in the ARCNET community because it was initially the only source of chips for ARCNET. Many boards are still made using chips supplied by SMC, although other sources and custom implementations are now on the market. SMC uses its own chip technology to build its own boards and hubs. SMC has also expanded ARCNET in two areas: twisted-pair wiring and multidrop (daisy-chain) coaxial cable. The nice thing about these alternatives is there is no cost difference for the boards, so it is simply a matter of choosing the more appropriate wiring method. These new versions of ARCNET use high-impedance drivers that differ from the low-impedance drivers found on most of the other ARCNET boards and the hubs from other sources.

The SMC PC-250 ARCNET card costs just under $300. The price is the same for a card with either coaxial or twisted-pair drivers. The SMC PC-500 is a 16-bit ARCNET card for under $700, with either coaxial or twisted-pair drivers. The coaxial version has a standard BNC connector. The twisted-pair version has a pair of standard RJ-11 telephone jacks. The coaxial version can be connected directly to an ARCNET hub, just as a normal ARCNET card could be. However, it is designed to be connected to other high-impedance ARCNET cards or hubs. In this case, the connection to the card is done using a BNC T-connector. The other ends of the connector are attached to coaxial cable, which is connected to the T-connector in the segment. Each end of a segment must have a terminator attached just like thin-wire Ethernet. The twisted-pair version works the same way, except that a T-connector is not required. Instead, each cable is plugged into the adjacent RJ-11 jacks of two computers and terminators, which are functionally identical to the coaxial versions, and are plugged into the cards at each end of the segment so all connectors have something plugged into them. A socket for a network boot ROM is included on all versions of the card.

The SMC PC-500 is a half-length card that is longer than the SMC PC-500. The SMC PC-500 adds card and cable activity LEDs and moves all switches to the top of the card. Unfortunately, the card is not downward-compatible from the device driver side, although new drivers are supplied for Advanced NetWare V2.0. Drivers for additional network operating systems such as VINES are being developed but were not available when the boards were being tested. The 16-bit card also doubles the on-board buffer size and adds a nodal priority feature that allows a card to gain access more often to the bus access token. This is useful for servers, which tend to send out a larger percentage of packets than do workstations. It can also be used to give high-performance workstations better throughput. Finally, the memory and I/O port selection is more flexible, which allows it to handle the AT/EGA/EMS problem.

The maximum segment length is only 400 feet versus the standard ARCNET limit of 2000 feet. However, high-impedance drivers effectively eliminate the need for hubs in small networks and reduce the number of hubs in larger networks. This is assuming that daisy-chain connections are suitable. In many cases, the PCs may be too far apart or may be in different rooms, which can make point-to-point connections, such as a normal ARCNET layout, preferable. The twisted-pair version of ARCNET can still be preferable in this case if the proper wiring is available or can be installed since telephone jacks can be found or installed almost anywhere. A common solution is to put hubs in the telephone wiring closet and use normal telephone wiring panels to connect the hubs to the proper telephone outlet. It is usually possible to use existing telephone wire, and two-or four-wire telephones can be available from the same outlet—using different wire pairs, of course.

SMC has two active, two-port external repeaters/hubs, which cost about the same as active four-port hubs from other sources. However, the proportional increase in cost versus the four-port, low-impedance hubs is usually offset by the increase in the number of PCs attached to a daisy-chained segment. One version of the two-port hub contains one coaxial connector and a pair of RJ-11 jacks. This type of box is used on mixed cable networks, with both coaxial and twisted-pair wiring. Either side of the network can include any number of adapters and repeaters. There is also no limit to the way coaxial and twisted-pair cable is used. A network could consist of a coaxial segment attached to the conversion repeater, attached to a twisted-pair segment, attached to another converter, attached to a coaxial segment, attached to a twisted-pair segment, and so on. An active eight-port external hub is also available for either the coaxial or twisted-pair cabling. The hub costs about the same as eight-port hubs from other sources but the SMC version has high-impedance drivers so each segment can be daisy-chained. All eight ports are of the same type, either twisted-pair or coaxial.

The external hubs contain a power LED but no activity LEDs. The adapter cards also lack both a line and card activity LED. SMC's documentation is very thorough, and drawings and examples are included throughout, making it one of the better boards to install. The adapter cards use switches and jumpers but they are not located on the back of the card, which means the network address can be changed only by removing the card. Although a network address is rarely changed, it can be annoying to remove the card to do so. The SMC card can change only the upper four address bits, which causes a problem on ATs with EGA and EMS boards. A special version of the card, which does let the card work with this combination, can be obtained from SMC upon request. The new version of the SMC card removes the problems by allowing the memory buffer to be placed appropriately using switches, and places the switches so that the card does not have to be removed to change the network address.

Thomas-Conrad Corporation

Thomas-Conrad provides premium cards at premium prices, but you get what you pay for. They are one of the few ARCNET suppliers with a 16-bit ARCNET adapter card. The TC6042 is the 8-bit, low-impedance adapter. The TC6045 and TC6046 are 16-bit versions for the AT and PS/2 MicroChannel bus, respectively. Because of its height, the TC6045 will fit only into a full-size AT chassis. It will not fit into an XT chassis, such as the IBM XT 286. The TC6042 and TC6045 are short cards. The TC6046 is a full-length PS/2 MicroChannel card. All three can operate in 8-bit compatibility mode while the 16-bit cards can be run in an enhanced mode with 16-bit data transfers. The enhanced 16-bit mode can provide additional throughput and reduced service overhead.

The cards have no monitor LEDs, and the interrupt used by the adapter is selected by a jumper on the XT and AT cards. The PS/2 card uses software selection as do most PS/2 MicroChannel cards. The AT card provides access to five additional interrupts, which can come in handy on servers with a large number of network or peripheral adapters. The memory and I/O address selections on the XT and AT cards are located on the lower part of the card, which means the card must be removed to change them. The network address selection switches are located near the top of the card and can be set with some care without removing the card, but the top of the PC must be removed. All cards contain a socket for an optional boot ROM.

High-impedance versions of the Thomas-Conrad ARCNET cards have the same limits and advantages as the SMC cards. The Thomas-Conrad hubs can also be used with the high-impedance cards.

The TC6040 is a passive four-port external hub. The next step up is the TC6050, which is an active eight-port external hub that can be upgraded to a 16-port hub. The active hub contains individual monitor LEDs for each connector.

The documentation is good. It describes adapter and hub setup as well as network layout and installation. The enhanced mode requires special drivers, and these are described in the documentation. A disk contains the additional drivers necessary for Novell NetWare. Contact Thomas-Conrad for availability of drivers for other network software.

Boot ROM

Most ARCNET cards support an optional boot ROM for diskless workstations. The ROM normally costs less than $50 and is available from different sources, depending on the net-

CHARTER SUBSCRIPTION INVITATION

Here At Last!
The Authoritative Source of Database Application and Product Information

Inside DBMS Magazine

DBMS Magazine offers in-depth articles examining the capabilities of programmable databases as tools for solving real business problems. Authoritative features show how a specific product is used to solve a specific problem. Sample some of the feature articles to be presented:

- 50 Nasty dBase Bugs, and How to Squash Them
- Has Ashton-Tate Finally Designed a Decent, Multiuser Database?
- Why 386 Databases Can be Either the Fastest, or a Waste of Money
- Is FoxBase+ dFastest Around?
- Why Microsoft and Ashton-Tate teamed up on SQL: Server – What this Ambitious Product Will and Will Not do for You
- R:Base Puts OS/2 to Work
- Understanding Paradox's PAL
- Can Oracle Squeeze Into a PC?

...with regular columns written by experts, providing tips on developing and modifying databases, such as:

- Putting dBase IV's Arrays to Work
- Cross tabulations in R:Base
- Sharing Files Between 1-2-3 and dBase
- Using Clipper's Valid Statement
- Tips on Creating Reports in Paradox

Who Should Read DBMS Magazine?

Professional developers, MIS/DP departments, consultants, VARs and serious users will find DBMS Magazine to be the source of important information for serious users of microcomputer database software.

Charter Offer and Benefits with No-Risk Guarantee

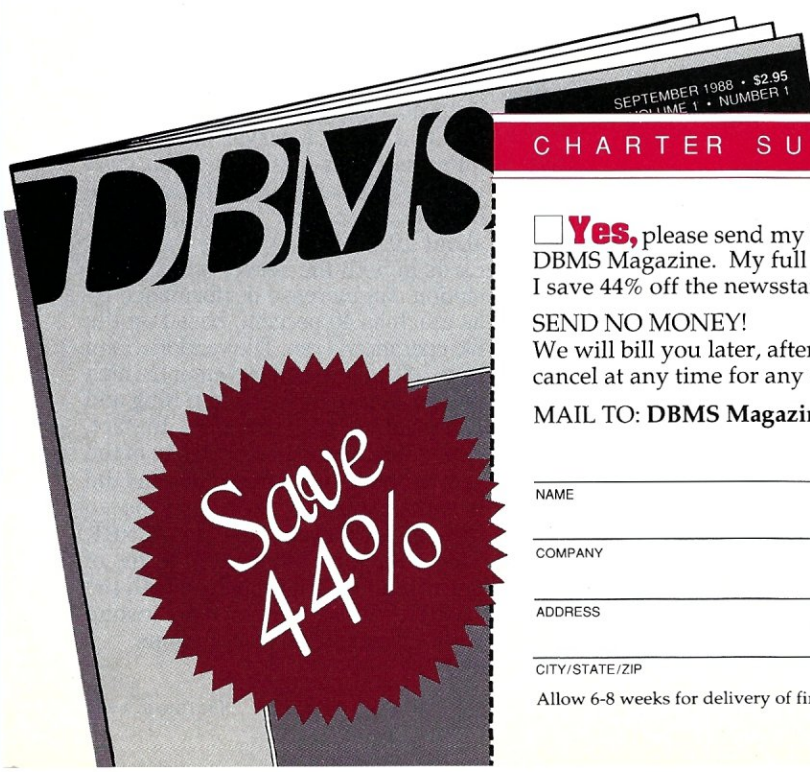
For a limited time DBMS Magazine will offer Charter Subscriptions with substantial cash savings.

You will also be entitled to continued savings — on all renewals and on any gift subscriptions you may order.

If at any time you decide DBMS is not for you, simply cancel. We will send a full refund for all unmailed issues.

Order Today!
Send your order form to:

DBMS Magazine
P.O.Box 3770
Escondido, CA
92025-9638



CHARTER SUBSCRIPTION ORDER FORM

☐ **Yes,** please send my first issue and enter my Charter Subscription to DBMS Magazine. My full year's subscription (11 additional issues) is only $19.97. I save 44% off the newsstand rate.

SEND NO MONEY!
We will bill you later, after you receive your first issue. GUARANTEE! You may cancel at any time for any reason and receive a full refund on all unmailed issues.

MAIL TO: **DBMS Magazine, P.O.Box 3770, Escondido, CA 92025-9638**

NAME

COMPANY

ADDRESS

CITY/STATE/ZIP

Allow 6-8 weeks for delivery of first issue. Offer good in the U.S. only. Newsstand rate (1 year): $35.40

20M8

work operating system software. Boot ROMs for Novell NetWare are the most common and are normally available from the vendors of the adapter cards. Any network software can theoretically support remote-boot workstations, but you will need to contact the vendor of the network software for details and availability of boot ROMs for ARCNET cards.

Workstations that use the boot ROM can still support floppy and hard disks. A NetWare boot ROM was tested on one of the workstations in the evaluation network. Installation of the network card was very easy for both workstations and the server —just set a few switches and jumpers and plug in the card. Adding the boot ROM support requires simply plugging in a ROM in the only open socket on the board. The boot ROM tested was from SMC. It checks for the availability of other boot devices and gives the option of booting from these devices, if possible. However, using a remote-boot option on a hard-disk system does seem to be a waste.

Fiber Optics and PS/2

Most of the companies that offer ARCNET cards are also delivering or developing ARCNET cards for the PS/2 MicroChannel as well as cards using fiber-optic drivers. Fiber-optic cards tend to be significantly more expensive than coaxial or twisted-pair cards and the fiber-optic cards run no faster than standard ARCNET cards. However, fiber-optic cards have a distance and noise immunity advantage. Hence, fiber-optic ARCNET cards or repeaters tend to be used in industrial areas and for networks that span multiple, adjacent buildings or very large buildings.

ARCNET Performance

All the 8-bit ARCNET cards perform the same, and they all seem to work together regardless of the cards or adapters being used. The ability to mix and match cards from different vendors is a great advantage. Mixing 8- and 16-bit cards works fine too. The 16-bit cards work in 8-bit compatibility mode and do not offer any noticeable improvement over 8-bit cards. Running the 16-bit cards in enhanced mode can improve performance depending on where they are installed. Putting the cards in a workstation and not in a server makes little difference, but putting a 16-bit card in the server and not in the workstations does improve performance by about 10 percent. Putting the 16-bit cards in both the server and a workstation can increase performance by as much as 40 percent, based on the file operations I ran. The vendors claim up to a 50-percent improvement, which brings it close to the Token Ring and Ethernet in performance. In fact, depending on the performance of the server, the server's hard disk, and the network software, there may be no difference between using ARCNET, Token Ring, or Ethernet in terms of performance. This was true with the 10-MHz AT server used in testing with AT and 80386 workstations.

Product Information

Earth Computer Technologies
10525 Lawson River Ave.
P.O. Box 8067
Fountain Valley, CA 92728
(714) 964-5784

4-port internal active PC hub	$ 400
8-port external active hub	695
Earthnet-I, ARCNETboard	199
Earthstation-I, diskless workstation	995

Circle reader service #254

Pure Data Inc.
1740 South I-35, Ste 140
Carrollton, TX 75006
(214) 242-2040

PDT508 Toshiba ARCNETboard	$ 595
PDI508 PC ARCNETboard	395
PDI508A 8-port external active hub	799
PDC508P 4-port internal passive hub	99
PDC504AX 4-port external active hub	495
PDC504AI 4-port internal active hub	395
PDI508F1 Half-Duplex Fiber Optic board	$ 995
PDI508F2 Full-Duplex Fiber Optic board	1295
PDC508AF1-1 Half-Duplex Fiber/coaxial Hub (1 fiber, 7 coaxial)	1395
PDC508AF1-2 Half-Duplex Fiber/coaxial Hub (2 fiber, 6 coaxial)	2395
PDC508AF2-1 Full-Duplex Fiber/coaxial Hub (1 fiber, 7 coaxial)	1695
PDC508AF2-2 Full-Duplex Fiber/coaxial Hub (2 fiber, 6 coaxial)	2695

Circle reader service #255

Standard Microsystems Corporation
35 Marcus Boulevard
Hauppauge, NY 11788
(516) 273-3100

PC-250 8-bit Controller Card	$ 295
PC-500 16-bit Controller Card	695
Diskless Workstation Boot ROM	50
2-Port Active Repeater	375
8-Port Active Hub	750
2-Port twisted-pair/coaxial Active Hub	375
PC-310 Fiber Optic Controller Card	1295
Active Hub-Opt (4 coaxial, 4 fiber)	4895
Active Link-Opt (1 coaxial, 1 fiber up to 4000 ft.)	1295
Active Link-Opt2 (1 coaxial, 1 fiber up to 11,500 ft.)	1495

Circle reader service #256

Thomas-Conrad Corporation
8403 Cross Park Dr., Ste. 1C
Austin, TX 78754
(800) 332-8683
(512) 836-1935

TC6042 ARC-CARD CE, 8-bit ARCNETcard	$ 289
TC6045 ARC-CARD/AT, 16-bit ARCNETcard	379
TC6046 ARC-CARD/MC, PS/2 ARCNETcard	395
TC6040 4-port passive hub	64
TC6050-10 8-port external active hub	695
TC6050-11 16-port external active hub	995

Circle reader service #257

The reason for the high performance of the 2.5 mbps ARCNET-based LAN is its design. ARCNET limits the number of stations that can be attached, which reduces the network message overhead by using a smaller address field. A small token also makes the rounds faster. This improves response time. Matching hard disk and cache to the number of workstations is important in a multiple workstation situation. The transfer rate of ARCNET is close to the actual transfer rate of hard disks found on many network servers, so overall throughput cannot exceed the hard-disk transfer rate, even if the network adapter can send data out faster. On the other hand, the use of RLL and ESDI drives plus 80386-based servers may require higher performance network adapters than ARCNET cards in order to provide optimal performance.

On the other side, ARCNET limits the network message to 512 bytes, including message overhead, which means a good amount of network traffic consists of multiple network messages. However, ARCNET has a fast turnaround, and it can run at full throughput with little overhead. Also, response messages tend to be small and do not have to be split into multiple packets. In most instances, delays or low throughput is due to hard-disk and network software performance.

So why not use ARCNET everywhere? Well, the comparison of ARCNET to Token Ring works well for LANs that directly connect fewer than 250 workstations, but Token Ring and Ethernet systems often have thousands of workstations on a single LAN and multiple servers. This is a different class of network from what ARCNET was intended for, but then again, there are a lot more LANs with fewer than 250 workstations and one or two servers than there are larger LANs. Also, multiple groups can be linked together by servers that are tied into multiple LANs.

Summary

ARCNET has most of the top features for LAN hardware: low cost, good performance at 2.5 mbps, efficient flow control using a token-passing protocol, and easy configuration. Twisted-pair support allows networks to be built using existing telephone wiring and patch boxes, while the coaxial version provides longer distances between hubs and adapters. The ability to mix and match can be a significant advantage, especially with existing ARCNET LANs. Overall, daisy-chain installations work best for a group of PCs in the same room, while PCs in different rooms or remote locations are usually wired using direct PC-to-hub connections.

ARCNET is a superior alternative to most lower speed solutions such as StarLAN. The IBM Token Ring works better for larger installations with almost twice the performance, but at a greater adapter cost. The Token Ring uses twisted-pair, but installation and maintenance are a bit more difficult than with ARCNET. Ethernet also costs more than ARCNET but, at 10 mbps, Ethernet does provide higher throughput that is on a par with the Token Ring. Don't underestimate ARCNET's robust configuration and installation procedures. Connecting and disconnecting adapters and hubs becomes more common as the size of the network increases.

All the cards tested worked together in the same network without any problems. The performance of the 8-bit cards was consistent, and it looks worth the money to put in 16-bit ARCNET adapters for file servers and high-performance 80386 machines. Earth Computer wins out in the price category, while SMC's high-impedance drivers allow the flexibility of using a daisy-chain configuration as well as the choice of twisted-pair wire versus coaxial. SMC's 16-bit card proves to be a very good performer. Pure Data's adapter for the Toshiba portable computer line is an inspiration. Thomas-Conrad leads the pack with a high price-performance ratio with 16-bit ARCNET adapters—performance-wise, it matched the SMC card. Both performed as well as Ethernet on the evaluation LAN.

ARCNET is a good solution for workgroups with fewer than 200 stations, which makes it suitable for almost any small- to medium-sized workgroup. ARCNET is also a good match for 80286-based servers and 16-MHz 80386 servers because they often do not have the horsepower to keep up with a lot of network requests so the servers are not always limited by the speed of the network adapters. There are alternatives to high-performance servers, including the use of 16-bit ARCNET cards or the insertion of multiple adapters in the server. Putting 16-bit adapters on high-performance workstations can also improve network performance. Fiber-optic alternatives provide a way to bridge distances between servers and workstations. □

Did you find this article particularly useful? Circle number 7 on the reader service card.

Heterogeneous Networking With NetWare

by Michael Hurwicz

Can Macintosh workstations share NetWare-based resources? This article explains how to connect different operating systems to Novell LANs.

Diverse PC networks are a fact of life for many companies. One department may have NetWare, another AppleTalk, and a third the IBM PC LAN Program. In general, companies would like to see a higher level of integration between these networks. For example, they would like to share files and printers among workstations on two or more dissimilar networks. Until recently, however, the major LAN vendors offered no means of integrating their LANs with those of their competitors. Systems integrators had to "go it alone" in connecting incompatible LANs, if indeed any meaningful connection was possible.

Now, however, Novell has announced a new Service Protocol Gateway Interface (SPGI), which provides a tool for network integration based on standard service protocols, such as the NetWare Core Protocol (NCP) used by NetWare, the Apple Filing Protocol (AFP) used by AppleTalk, and the Server Message Block (SMB) protocol used by MS-NET networks such as the IBM PC LAN Program and 3Com's 3+. Although all the details of Novell's SPGI have not yet been released, its basic principle is simple: A Service Protocol Gateway (SPG) machine which is connected to two networks that use dissimilar service protocols, translates from one service protocol to another. This is possible since the service protocols are functionally similar in many respects. For example, they typically include functions for opening and closing files and for locking and unlocking portions of files.

Like human languages, service protocols are close enough so that rough equivalences can be set up between them. Something may be lost in the translation, and it may be impossible to translate some functions, but a great deal can be done that is useful. For example, the first SPG to be introduced by Novell links AppleTalk networks with NetWare file servers, thus allowing workstations on the AppleTalk LAN to see the NetWare file server as an AppleShare server.

SPG Requirements

The SPG runs as a NetWare Value-Added Process (VAP,) which is the mechanism provided by Novell to allow third parties to write server-based or bridge-based applications in the NetWare environment. This has at least two implications: First, the SPG cannot exist by itself but must run in a machine with NetWare (only NetWare file servers or NetWare bridges can run VAPs and thus host SPGs).

Michael Hurwicz, a writer and consultant specializing in local area networks, is based in Nashville, Tennessee.

Simply the BEST C and Pascal on AT, 386, Sun, Apollo, RT, VAX, 370

"The most rock-solid C compiler in the industry. Superb technical support and portability. Superior code generated."
Gordon Eubanks, Symantec — Q&A (386).

"It simply works, with no trouble, no chasing strange bugs, and excellent warning and error messages ... a professional product."
Robert Lerche, Bay Partners.

"For large-scale software development, the highest quality C compiler available on the market today. Pragmas are great. Quality of support is exceptional." **Randy Neilsen, Ansa—Paradox (DOS,OS/2).**

"15% smaller and 15% faster than Lattice C."
Robert Wenig, Autodesk.

"Our software is running anywhere from 30 to 50% faster than when compiled under Lattice." **David Marcus, Micronetics.**

"We switched from Lattice due to a 10% reduction in code size. The compiler is very stable." **Lee Lorenzen, Ventura Software — Ventura Publisher, marketted by Xerox Corp.**

"Best quality emitted code by any compiler I've encountered. Often amazing." **Bill Ferguson, Fox Software — FoxBase (386).**

"Messages sometimes pointed out type mismatches, incorrect-length argument lists, and uninitialized variables that had been undetected for years [in 4.x bsd]." **Larry Breed, IBM ACIS [RT PC].**

"Diagnostics turned up bugs missed by other compilers. Rapid bug fixes by technical support, someone who knew what he was talking about. 80386 code is well optimized."
Tim Addison, Logistics Data Systems.

"386 protected mode support is fantastic, especially the access to large amounts of memory. It's mainframe compute power on a PC." **Dan Eggleston, Viewlogic.**

"The preprocessor supplied with Professional Pascal is quite useful. The code quality and control over segmentation and memory models are superior to MS Pascal." **Bob Wallace, QuickSoft.**

Check Out These Reviews

- **High C ™:**

Computer Language	February 1986, '87	
Dr. Dobb's Journal	August 1986	
PC Magazine	Jan. 27, 1987	(80386 version)
Dr. Dobb's Journal	July 1987	(80386 version)
BYTE Magazine	November 1987	(80386 version)

- **Professional Pascal ™:**

PC Magazine	Dec. 29, 1985	
Computer Language	May 1986	
PC Tech Journal	July 1986	
Journal of Pascal, Ada, & Modula-2	Nov.-Dec. 1986	
BYTE Magazine	Dec. '86, June '87	(80386 version)

Why MetaWare compilers

- They are specifically designed for serious software developers.
- They are reliable and robust: they don't break at every turn.
- Their generated code is the best, or near best, on each architecture.
- Their superior diagnostic messages help you produce better products more quickly.
- Your source can be ported with ease to the most popular systems.
- You can link mixed-language modules from our compilers, others
- You can benefit from high-level, personal technical support.
- You can take advantage of the latest ANSI C extensions, and/or extensive Pascal extensions. **High C** has been tracking the ANSI Standard for two years; **Professional Pascal** will soon have a VS Pascal compatibility switch and several Apollo Pascal ext'ns.
- You can take advantage of the latest 387 and Weitek 1167 support — we have the only compilers with Weitek real mode support.

Power Tools for Power Users

Ashton-Tate: dBase III Plus, MultiMate; Autodesk: AUTOCAD, AUTOSKETCH (8087, '387, Weitek); Boeing Computer Services (Sun); CASE Technology (Sun); CAD/CAM giant Daisy ('86, '386, VAX); Deloitte Haskins & Sells; Digital Research: FlexOS; GE; IBM: 4.3/RT, 4680 OS; Lifetree Software (Pascal): Volkswriter Deluxe, GEM-Write; Lugaru: Epsilon; NYU: Ada-Ed cmplr; Semantec: Q&A; Sky Computers; ... (Product names are trademarks of the companies indicated.)

Industrial-Strength

MetaWare C and Pascal compilers are designed for professional software developers. These tools are loaded with options to control them for special purposes. You can adjust the space-time trade-off in code quality. You can adjust external naming conventions to agree with linkers and operating systems. You can specify segment names for segmented architectures, and to help place code or data in particular places for embedded applications. You can select from five memory models for the 8086 family. And on and on.

A Partial List of Optimizations

Common subexpression and dead-code elimination, retention and reuse of register contents, jump-instruction size minimization, tail merging (cross jumping), constant folding, short-circuit evaluation of Boolean expressions, strength reductions, fast procedure calls, automatic mapping of variables to registers (where advantageous), ...

"Platform" — Code Quality

Sun,Apollo,SGI	— 18%, 3%, 26% > resident compiler (Dhrystone).
PC: DOS, OS/2	— 3-10% > Microsoft C; 30% > MS Pascal, LatticeC.
386 32-bit DOS	— no competitors, since November, 1986.
286, 386 UNIX	— 66% better than pcc (Dhrystone, 386).
VAX VMS	— ≈ DEC's excellent C and Pascal; better features.
VAX Ultrix	— 19% > pcc (Dhrystone); much > Berkeley Pascal.
RT PC/4.3bsd	— 89% > the original port of pcc (Dhrystone).
370 CMS,UNIX	— much better than any C, and VS Pascal.
AMD 29000	— >40,000 Dhrystones! Available in Q2, cross.

(408) 429-6382, telex 493-0879. ***Since 1979.***

Meta Ware™
INCORPORATED
903 Pacific Avenue, Suite 201 • Santa Cruz, CA 95060-4429

The Clear Choice for Large Programming Projects — PC Tech J.

Second, SPGs require NetWare Version 2.1 or later since versions released prior to that do not support VAPs. In fact, particular SPGs may require the latest version available at the time the SPG is released.

Advantages and Disadvantages

The gateway approach to network integration has at least three advantages: (1) It is nonintrusive; (2) It is relatively independent of everything except the service protocols themselves; and (3) It is easy to manage. It also has three disadvantages: (1) It may not be as fast as some alternative approaches; (2) It takes up a significant amount of memory in the server or bridge; and (3) It may impair performance of other applications using the server or bridge.

First, the SPG approach has the advantage of being nonintrusive. Existing software, whether it be applications, workstation operating systems, or network operating systems, continue to function as usual, unaffected by the translation process that takes place. Thus, customers' investment in LAN software and hardware is protected. Nothing has to be thrown away. And users don't have to adapt to a radically new environment.

It may be possible and appropriate to create new NetWare-specific utilities that run in workstations on the client network. Novell has created such utilities for the Macintosh gateway. However, these are standard applications running under the Macintosh Operating System, not special networking modules. In particular, Novell did not have to implement a NetWare shell for the Macintosh. (The NetWare shell is the software that interfaces a NetWare workstation to the network.)

A final advantage of the nonintrusive approach is that any problems with the gateway affect only the gateway and the applications that use it. The chances for unwanted side effects in operating systems or other applications are minimized.

The use of standard protocols provides independence from numerous factors that may change over time or from installation to installation. The standard protocols themselves, though they may be extended, are seldom changed. Hardware, applications, workstation operating systems, and network operating systems, on the other hand, can

> *The SPG approach has the advantage of being nonintrusive. Existing software . . . continue to function as usual.*

change radically. As an example of this benefit, SMB packets generated under OS/2 are indistinguishable from SMB packets generated under DOS. Presumably, the same SMB protocol will be used in whatever workstation operating system provides the basis for the next generation of MS-NET. Thus, if an SPG were created for MS-NET, for instance, the same SPG would be able to handle both OS/2 and DOS workstations, and, presumably, future MS-NET workstations as well.

Gateways are also easier to manage than software that runs in every workstation. For instance, when SPG software is updated, the LAN manager doesn't have to make sure that the new software is installed on each workstation. The software is installed just once, at the server or bridge. Similarly, any problems will be localized at the server or bridge.

On the downside, a protocol-based gateway is probably not the fastest way to link networks. Had Novell implemented the NetWare shell for the Macintosh, for instance, each workstation could format NCP packets and perform other work, such as checking security rights. With the gateway approach, the gateway has to perform all these functions. It is reasonable to suppose that offloading those tasks from the gateway and distributing them to a number of workstations would speed things up. (Novell conducted some tests, however, which indicated that the impact would not be great. Apparently, the workstation-based approach generates more LAN traffic, largely canceling the gain from distributing the processing.)

SPGs will also take up a significant amount of memory in the server or bridge. On a file server, the memory consumed by the SPG takes away from memory that could be used for disk caching and other buffering. This becomes critical on a bridge. In contrast to NetWare servers, which run in protected mode and can support large amounts of memory (typically up to 16 MB), NetWare bridging software runs only in real mode, limiting memory to 640K. Thus, it may not be possible to run several SPGs in a bridge, or to run an SPG and another VAP. For example, although Novell is offering a print services VAP with the Macintosh gateway, the two cannot run concurrently in the same bridge.

Finally, the SPG may impair performance of other applications using the server or bridge. This is actually a generic problem with all VAPs and more broadly found in most multitasking environments. At some point, the processor becomes overloaded, and performance is hurt for all tasks. This is, of course, an unpredictable effect that depends on how heavily the SPG is used and how much processor time the other tasks require. In many typical environments, other tasks may see no appreciable degradation in performance.

Creating a Gateway

Physically, an SPG is a file server or bridge with two, three, or four different network interface cards: one or more for the client network and one or more for the NetWare network. Currently, the architecture calls for the client system to have its own hardware in the gateway machine. For the Macintosh gateway, however, Novell has changed its Ethernet drivers so that NetWare LANs and AppleTalk LANs can share the same LAN card in the gateway machine, as long as a specific 3Com LAN card is used. A generic approach to such sharing is a future possibility.

SPG software consists of three main components: a platform, a client half, and a server half. The relationship of the three parts is as follows:

1. The platform contains some very basic calls that both of the other components need in order to get started and spawn processes.
2. The client half accepts requests from workstations on the non-NetWare LAN and translates those requests into SPG calls.
3. The server half accepts the SPG calls and translates them into NCP calls.

4. The client and server halves communicate using the Service Protocol Gateway Interface.
5. The client network interface card requires a hardware driver. (The hardware driver for the server half is provided by Novell.)

A developer outside Novell will typically create the client half and the hardware driver for the client hardware. Novell provides the platform, the server half, and sample code for the client half. For the Macintosh gateway, for instance, Dayna Communications of Salt Lake City, Utah, created the client half, implementing the Apple Filing Protocol and hardware drivers for the LocalTalk and EtherTalk hardware, while Novell created the server half.

In operation, each half of the SPG runs as a separate set of processes. In the NetWare environment, when a process receives control of the processor it can keep control as long as it wants. It is therefore important that the SPG halves be "nice guys" and do their processing in small, efficient chunks. Typically, the client half receives a request from its network, performs the necessary translation, calls a routine in the server half which queues the request, and then relinquishes control. When the server process receives control, it processes the request from its queue, calls a routine in the client half to queue the response, and then relinquishes control. In this way, the client and server halves operate quickly and independently.

> *Almost all SPG functions are requested through function calls compatible with the C programming language.*

It's Not That Easy

As one might suspect, the actual creation and use of an SPG is somewhat more complex than this simple description might suggest. Most of the complexities arise from the fact that the two service protocols involved cannot be matched up detail for detail. For instance, NCP security is a good deal more sophisticated than that of AFP. When AFP clients access the NetWare file server, they are subject to NetWare security rules, but they use Finder-compatible tools to manipulate that security. This can lead to some interesting results.

For instance, AFP combines "open," "create," "modify," "write," and "delete" directory access rights into one right, the "Make Changes" privilege. It is not possible to allow an AFP user to write to files in a particular folder (directory) while withholding the right to delete files in that folder, for example. NCP, in contrast, separates these rights. However, whether the AFP user has one, two, or several of these rights, the Finder icon for the folder and the Finder icon displayed in the upper left-hand corner of the window of an open folder both look the same, and the "Get Privileges" option under the File menu results

ADVERTISER INDEX

ADVERTISING REPRESENTATIVES

National Account Manager Dwight Schwab (415) 366-3600
National Account Manager Tami Brenton (415) 866-1957
Advertising Coordinator Shaun Hooper (415) 366-3600

Associate Publisher
Michael Wiener

in the same listing.

All these indications would lead the AFP user to believe that he or she can write to or delete NCP files in that folder. However, if the user attempts a forbidden operation, an "access rights violation" message results. For example, the user might have write privileges but not delete privileges in the directory. In that case, attempting to delete a file would result in the error message—a potentially puzzling occurrence for the Macintosh user who thinks that he or she has the right to make changes in the directory.

Programming for the SPG

In general, VAP programming is complex. Novell has tried to make SPG programming a little easier. Since SPGs have specific and limited requirements, Novell created the SPG Platform to provide just those functions required by client halves. For instance, VAPs normally have access to a variety of low-level network and transport calls used mostly for workstation-to-workstation communication. All of these calls are not needed by an SPG client half. As another example, VAPs can use a Service Advertising Protocol (SAP) to announce their presence on the network. SPGs do not announce their presence in this way, so the SAP interface is not supported by the SPG Platform.

Almost all SPG functions are requested through function calls compatible with the C programming language. (Some functions connected with initialization may be requested using assembly language routines.) The SPG client half first creates an "SPG Request Block" and then calls the "Service Request" function, passing to the function a pointer to the request block. The server half uses the pointer to find the request block and responds to the request.

Listing 1 shows a typical Service Protocol Gateway Interface request block implemented in C.

Listing 2 shows how the client makes a service request to the server half. This example assumes that the client half has the *servername*, *username*, and *password* in appropriate locations. The first call establishes a new connection. At the time the connection is established, the server half fills in the fields of the *ServiceConnectionNumber*. The client will use this information in all future calls from the same user.

The *RequestService* function returns immediately. The server half simply queues the request. When the service has been performed the server half will call the client routine *ServiceRequestDone*. *GetNewConnection* initializes the *ClientServiceConnection* in the request block. Typically, a request block would be allocated for each user.

Listing 3 shows how the client would call *ReadFile*. It assumes that the client has obtained a valid file handle

Listing 1. Typical Service Protocol Gateway Interface request block.

```
/* Typical declaration of a SPGI request block */

/* First a structure for POINTER parameters. POINTER is an eight-
byte structure with following layout:
  2 bytes:  Client length.  Number of bytes of data passed to
     server half by client half, or maximum length of
     buffer which may be filled in by server half.
  2 bytes:  Server length.  Actual number of bytes filled in by
     server half.
  4 bytes:  Address of buffer. */

struct     PtrStruct {
                int  CLength;            /*Client length*/
                int  SLength;            /*Server length*/
                char far  *BufferAddress
};


/* Now actual Request Block */

struct     SPGIRequestBlock     {
                void far       *Link;

/* "link" field is needed by server half to manage request
queues. While client half has control of block, it may
use link field as it wishes. */

                int            RequestClass;
                int            RequestType;
                int            ServiceConnectionNumber;
                int            TaskNumber;
                unsigned int   ErrorCode;
                char           ErrorClass;
                char           ErrorLocus;
                char           ErrorAction;
                char           NotUsed;
                struct PtrStruct     PtrParm[10];
                int            WordParm[10];
                long           LongParm[10];
};
```

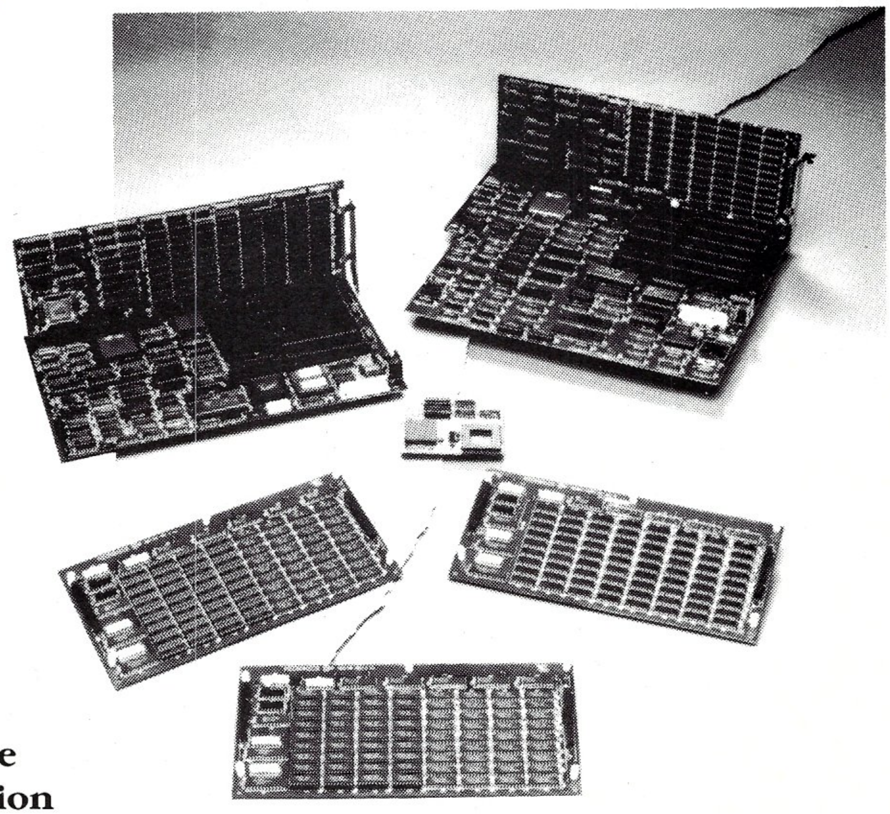
Listing 2. How client makes a service request to the server half.

```
/* First, set up variables, arrays and request block structure
required by "Get New Connection" function call, and declare
function itself */

struct     SPGIRequestBlock     RequestBlock;
char       ServerName[32];
     /*  Assume data is "Accounting" which is 10 bytes */
char       UserName[32];
     /*  Assume data is "Marilyn" which is 7 bytes */
char       UserPassword[32];
     /*  Assume data is "poiuyt" which is 6 bytes */

     extern    far  ServiceRequest();


     ...

/* Then call GetNewConnection which is class 0 type 0 */

     RequestBlock.RequestClass = 0;
     RequestBlock.RequestType = 0;
     RequestBlock.TaskNumber = 0;

/* Place appropriate values in request block. Note that
client indicates actual length of client data and not maximum
length of buffer in this call.                     */

     RequestBlock.PtrParm[0].CLength = 10;
     RequestBlock.PtrParm[0].BufferAddress = &ServerName[0];

     RequestBlock.PtrParm[1].CLength = 7;
     RequestBlock.PtrParm[1].BufferAddress = &UserName[0];

     RequestBlock.PtrParm[2].CLength = 6;
     RequestBlock.PtrParm[2].BufferAddress = &UserPassword[0];

/* Request the service. */

     RequestService(&RequestBlock);
     ...
```

Listing 3. How client would call ReadFile.

```
/* Define necessary variables and arrays. */

     long FileHandle;
     long FilePosition;
     char DataBuffer[512];
     /*  Enough space for largest data request */

     ...

/* Place appropriate values in request block. */

     RequestBlock.RequestClass = 0;
     RequestBlock.RequestType = 5;
     RequestBlock.LongParm[0] = FileHandle;
          /*obtained from an OpenFile request */
     RequestBlock.LongParm[1] = FilePosition;
          /* Determined independently */
     RequestBlock.PtrParm[0].CLength = 27;
          /* Requests 27 bytes of data */
     RequestBlock.PtrParm[0].BufferAddress = DataBuffer;

/* Request the service. */

     ServiceRequest(&RequestBlock);
     ...
```

Insist On The Best Micronics 386 Motherboards



Quality,
Performance
and **Innovation**
best describe our 80386 based board level product line. Now with both **AT** and **Baby size** and high speed **CACHE** memory. Micronics is the leading supplier to **OEMs, VARs** and Systems Integrators that require the **best in 80386 technology.**

For a distributor near you, call
800 / 234-4386.

Innovation
and
Performance

MICRONICS
COMPUTERS INC.

CIRCLE 113 ON READER SERVICE CARD.

and that the *ClientServiceConnection* in the request block has been initialized.

When control returns at *ServiceRequestDone*, the server half will have transferred up to 27 bytes of data into *DataBuffer* and set *RequestBlock.PtrParm[0].SLength* to the actual number of bytes transferred.

Novell provides object files for the portions of the gateway that it provides—namely, the platform and the server half. The developer creates and compiles the client half and the client hardware driver. All these pieces are then linked together to make the gateway.

> *The introduction of service protocol gateway technology may herald an age which offers more control over the LAN environment*

SPG Calls

The Service Protocol Gateway Interface contains five classes of calls: SPG platform calls, connection and basic I/O calls, file and directory calls, miscellaneous NCP calls, and client-specific calls.

1. SPGplatform calls are divided into three areas of functionality: VAP interface calls, client platform calls, and server platform calls.

    - The VAP interface calls perform three basic types of functions: memory allocation, process control, and initialization. These calls allow the program to allocate memory, start running, and spawn processes.
    - There are just two client platform calls: "Client start" and "Client down handler." The Novell-provided portion of the SPG software calls "Client start" to start the client process. The Novell-provided portion of the SPG software calls "Client down handler" to terminate the client. This may be necessary, for example, when the server is down; calling the down handler gives the client half time to flush all its buffers and perform whatever cleanup is necessary before termination.
    - The Novell-provided portion of the SPG software also calls the "Server start" and "Server down handler" functions, which parallel the client platform calls.

2. Connection and basic I/O calls, also referred to as "Class Zero" calls, allow the SPG to establish and delete connections and perform basic file I/O functions such as creating, opening, reading, writing, and closing files.
3. File and directory calls, also called "Class One" calls, include:
    - File management, including deleting and renaming files.
    - Directory management, including making, renaming, searching, and removing directories.
    - Record-locking.
    - An "end of task" call.
4. Miscellaneous NCP or "Class Two" calls are NetWare-specific calls. For instance, this class of calls manages security on the NetWare server.
5. Client-specific calls are categorized as "Class Three" or greater. As the name implies, these are calls required by a specific type of client but not by other clients. For instance, AFP files have a "data fork" and a "resource fork," while PC files do not. Thus, for the Macintosh gateway, Novell created *MacOpenFork*, a call that allows programs to open a data fork or resource fork. In order to implement this, Novell actually had to add a new function to NetWare. Such functions, incidentally, will be available to all NetWare workstations in future versions of NetWare. The existence of such new functions explains why a particular gateway may require the latest version of NetWare in order to operate. These calls will be added to NetWare in conjunction with developing the gateway itself.

Summing Up

Connecting microcomputer LANs from different vendors has been like the weather: Easy to talk about, but hard to do anything about. The introduction of service protocol gateway technology may herald the coming of an age in which systems integrators can offer LAN users more control over the LAN environment. □

Did you find this article particularly useful?
Circle number 9 on the reader service card

All the source code for articles published in Micro/Systems *is available on MS-DOS disk. To order, send $14.95 to* Micro/Systems Journal, *501 Galveston Drive, Redwood City, CA 94063; or call Tim at (415) 366-3600. Please specify the issue number. Source code is also available on CompuServe; type GO DDJFORUM and check the data library, line 14.*

Product Information

Novell, Inc.
122 East 1700 South
Provo, UT 84601
(801) 379-5900
Circle reader service #259

Apple Computer, Inc.
700 Laurelwood Road
Santa Clara, CA 95054
(408) 748-9303
Circle reader service #260

Dayna Communications
50 South Main St., Ste. 530
Salt Lake City, UT 84144
(801) 531-0600
Circle reader service #261

Writing Batch Files for Easy LAN Directory Access

by Brian Brown

These two batch-file utilities make it a snap to pop in and out of various network directories.

I work in a large, technical college that uses several PC-based networks (all using Novell NetWare). Often, programs that reside on the network must be invoked by a batch file, which changes the current drive and directory to that of the application program. A typical batch file might look like:

```
ECHO OFF
F:
CD \USR\TURBO
TC
```

This batch file changes the workstation access to the network drive, and runs TURBOC. There is, however, one problem associated with batch files of this nature. When the user is finished running the application (in this instance TC), they are not returned to the drive and directory that they were in before running the batch file.

Batch files such as the example above have been implemented because some applications access auxiliary files (overlays, on-line help, etc.) when the application is run. Often, these auxiliary files must be in the current directory when the application is invoked. Some recently introduced packages allow users to specify where these might be found, others packages don't.

Which, of course, brings us to what this article is all about. Wouldn't it be nice to have a batch file that looks like this?

```
ECHO OFF
PUSHCD
F:
CD \USR\TURBO
TC
POPCD
```

The program PUSHCD saves the current drive and path name (directory). The application is then invoked and, upon return the program, POPCD restores the old drive and directory. Voila, you are back where you started from. And it not only works as a utility on stand-alone ATs and PCs, but it also works on networks (as long as each user has a unique user name or user-id).

How, then, do these programs work? Let's look at PUSHCD first (Listing 1). It is written in Turbo C and is compiled to a stand-alone .EXE file. The technique involves using the DOS calls ***getcwd( )*** and ***getdrive( )*** to obtain the current directory and drive, then saving these in a specified file. POPCD (Listing 2) reverses the operation by obtaining the drive and path name from the specified file, then using the DOS calls ***setdisk( )*** and ***chdir( )*** to restore the user to the original starting point.

These utilities also can be used separately from batch files. As long as they reside in the specified search list (set up by the PATH command), they can be invoked from any directory. A typical use might be:

```
A:\INTRO>pushcd
A>C:
C:>cd . . . .
. . . . .
C:>popcd A:\INTRO>
```

This batch file allows you to save where you are, buzz off and work somewhere else, then jump back where you started by typing POPCD.

Using PUSHCD and POPCD in a Network

However, things are a bit more complicated when transferring this concept to a network environment. For one thing, the file created and used by the programs PUSHCD

Brian Brown is a lecturer in Software Engineering at Central Institute of Technology, Wellington, New Zealand.

and POPCD would need to be different for each user. Otherwise, a user will probably come back somewhere else because another user has over-written the saved file information.

What we really need is a way in which each user can create a unique file name. This technique will work only if each user has a different login name. Under Novell NetWare Version 2.x, you can do the following in each users login script:

```
SET usr=LOGIN_NAME
```

When a user, such as BRIAN, logs in to the network, this variable LOGIN_NAME is replaced by the words BRIAN and inserted in the DOS enviroment area. Simply put, once logged in, type the DOS command SET to display the environment variables.

```
A>SET USR=BRIAN
```

This is great stuff because if PUSHCD and POPCD are changed to access the DOS environment and extract the USR name, which will be different for each user, this name can be used to create separate files for each user. Turbo C provides the function *getenv( )* to access the environment table. Listing 3 shows a stand-alone program that will list the user name (as long as the login script has been changed to support this).

Now it is relatively simple to change the programs PUSHCD and POPCD so they use unique file names. Listing 4 shows the alterations necessary for PUSHCD.

Potential Pitfalls

Well, there is no gain without pain, so what are the problems you have to watch out for?

First, be sure that each user has a unique login name, and remember to modify the login script to include the USR variable.

Second, the programs do not support recursion. In other words, don't go zapping off all over the place using PUSHCD each time, then expect to return to the starting point using POPCD. The POPCD utility will only return you to the last saved drive and directory.

Properly used, however, PUSHCD and POPCD could add a little more flexibility to machine and network management than you might have at present. □

Did you find this article particularly useful?
Circle number 13 on the reader service card.

Listing 1.

```
/* Pushcd.C
    Save current drive and directory
     in c:\usr\tmp\orgcd.tmp */

#include <dir.h> /*prototypes for getcwd() & getdisk()*/
#include <stdio.h>

main()
{
  int drive;
  char path[40];
  FILE *fout, *fopen();

  drive = getdisk();
  if( getcwd( path, 40 ) == NULL )
  {
   printf("\nPushcd: Error, can't get full pathname.\n");
   exit(1);
  }
  fout = fopen( "c:\\user\\tmp\\orgcd.tmp", "w" );
  if( fout == NULL )
  {
   printf("\nPushcd: Error, can't create temp file.\n");
   exit(2);
  }
  fprintf( fout, "%d\n%s", drive, path+2 );
  fclose( fout );
}
```

Listing 2.

```
/* Popcd.c
    Restore old drive and directory
     from c:\usr\tmp\orgcd.tmp */

#include <dir.h> /*prototypes for cddir() & setdisk()*/
#include <stdio.h>

main()
{
  int drive;
  char path[40];
  FILE *fin, *fopen();

  fin = fopen( "c:\\user\\tmp\\orgcd.tmp", "r" );
  if( fin == NULL )
  {
    printf("\nPopcd: Error, temp file does
                   not exist.\n");
    exit(1);
  }
  fscanf( fin, "%d", drive );
  fscanf( fin, "%s", path );
  fclose( fin );

  setdisk( drive );
  if( chdir( path ) == -1 )
  {
    printf("\npopcd: Error, can't change directory.\n");
    exit(2);
  }
}
```

Listing 3.

```
/* getenv.c  demo program to get username variable */

#include <stdlib.h>
#include <stdio.h>

main()
{
  char user[20], *username;

  username = &user[0];
  username = getenv( "USR" );
  printf("\nUsername is %s\n", username );
}
```

Listing 4.

```
/* Pushcdn.c
     Save current drive & directory in
      c:\usr\tmp\??????.tmp */

#include <dir.h> /*prototypes for getcwd() & getdisk()*/
#include <stdio.h>

main()
{
  int drive;
  char filename[12], path[40], user[20], *username;
  FILE *fout, *fopen();

  username = &user[0];

  drive = getdisk();
  if( getcwd( path, 40 ) == NULL )
  {
    printf("\nPushcd: Error, can't get full pathname.\n");
    exit(1);
  }

  username = getenv("USR");
  strcpy( filename, "c:/user/tmp/");
  strcat( filename, username );

  fout = fopen( filename, "w" );
  if( fout == NULL )
  {
    printf("\nPushcd: Error, can't create temp file.\n");
    exit(2);
  }
  fprintf( fout, "%d\n%s", drive, path+2 );
  fclose( fout );
}
```

Transmitting PostScript Files From DOS

by Thomas A. Dwyer

How do you use an Apple LaserWriter from a DOS environment? The first step is to create a PostScript bridge.

The debate over the DOS versus Macintosh style of computing is based on a number of factors, most of them software-related. The perceived value of such software may very well depend, in turn, on how fluently the software interfaces with state-of-the-art hardware (e.g., a laser printer).

This connection with special hardware is particularly important in desktop publishing. Users of MacWrite, for example, love the software for its simplicity, and for the way in which its output can be sent to an Apple LaserWriter to produce striking visual results. However, many professional writers and publishers feel that the real winner in terms of desktop publishing software is the DOS-based Ventura Publisher—and part of this assessment, at least, is due to its ability to access high-quality printers. [A good discussion of the pros and cons of Ventura, and the problems involved in accessing high-quality output devices, can be found in Dave Thompson's articles in *MicroCornucopia* (September/October and November/December 1987) where he describes the transition of his magazine to a Ventura-produced product.]

It's not difficult to become a Ventura enthusiast. Its impressive sophistication and power are enough to gladden the hearts of even the most jaded computer users. Ventura also supports the Apple LaserWriter (and most other PostScript printers). For users who don't own a PostScript printer, the question of how to use a remote laser printer then arises.

The DOS-To-LaserWriter Connection

The method I have settled on for my own use is simple and reliable. It makes use of the fact that one of the many nice features of Ventura is that you can tell it you have a LaserWriter, and then correct for that little fib by actually sending your printer output to a disk file. (This same capability is also starting to appear in word processors, and I have successfully used the techniques described here with WordPerfect.)

The disk file you create via this subterfuge will be a PostScript version of your document. You can then walk the disk to the nearest friendly PostScript-speaking printer and try to feed it your PostScript file. If the printer is not within walking distance but is connected to a computer with communication capabilities, you can send the file via a phone connection and two modems, or possibly via a

---

Thomas Dwyer is Professor of Computer Science at the University of Pittsburgh, with interests in microcomputing, computer graphics, and the application of computers to education.

network that both machines can access. This latter option has complications of its own, however, which I won't discuss further except to mention that I have succeeded in sending DOS Ventura files to the DEC PrintServer-40 PostScript printers on our campus network. (One tip: Remove all Ctrl-D's from your file before sending it via a mainframe host.)

Getting back to use of the "walk-net" approach: When you arrive at the printer location, you'll be faced with two possibilities. The first is that the LaserWriter is connected to a Mac, in which case your DOS disk is of little use (even if it is 3½ inches in diameter!). Let me skip over this problem for a moment; as we'll see below, it's surmountable, but you'll need to invest in some additional hardware.

The second option is that the printer is connected to a DOS machine that reads your disk format. In this case, the picture is a lot brighter. Unfortunately, following the instructions that come with the LaserWriter (found in an appendix that grudgingly mentions DOS machines) doesn't quite work. It can be done, however, and relatively easily. Here's the formula I've come up with:

1. If necessary, construct a cable that connects the COM1 port on the DOS machine to the 25-pin RS-232 socket on the LaserWriter. The cable should carry lines 3 and 7 (ground) straight through, but cross over lines 1 and 2; that is, connect the wire from pin 1 at one end of the cable to pin 2 at the opposite end, and vice-versa. (You do this because the LaserWriter and COM1 are both configured as DTE devices.) At the PC COM1 end, connect pins 5 (CTS) and 6 (DSR) together, and connect a wire from this junction over to pin 20 (DTR) at the LaserWriter end of the cable. Alternatively, if you happen to have a modem eliminator cable, you can use this instead, although that's a bit of overkill.
2. Set the selector switch on the LaserWriter for 9,600-baud serial communication.
3. Set the COM1 port communication parameters as follows:

   ```
   A>mode com1:9600,n,8,1,p
   ```

   (Note: The manual for the Apple LaserWriter II incorrectly states that the parameters are 9600,n,7,1,p. This won't work.)
4. Assure yourself that the LaserWriter is operating with its normal XON/XOFF protocol, which is the way it is usually configured. Unless someone has changed to the optional DTR protocol, there's probably nothing for you to do at this step. However, the Apple manuals do have programs for changing to the DTR protocol, and if this has been done, you'll want to change back to XON/XOFF. Simply send the PostScript program XON.TXT shown in Listing 1 to the printer with the copy command (*copy XON.TXT com1*). It is recommended that the printer then be left in this configuration, since changes are made in EEROM,

Listing 1. XON.TXT

```
% Use DOS copy command to send this file via com1 to
% LaserWriter II-NT.  This will set flow control protocol to
% XON/XOFF.  Since this is a PostScript program, it should begin
% and end with a control-D character.
%
serverdict begin 0 exitserver
statusdict begin
revision 2 ge {25 sccbatch exch pop 0 ne {25 9600 0 setsccbatch} if}
if defaulttimeouts 60 lt {60 setdefaulttimeouts}{pop pop}ifelse
```

and the protocol can be changed only a finite number of times. Once the change has been made, the XON/XOFF setting will remain, even after power is removed. Since XON/XOFF is the protocol used in the Mac world, and since the program presented in Listing 2 of this article makes XON/XOFF transfer available to any DOS machine, there is really no reason to ever switch to DTR, as suggested in the Apple manuals.

5. You're now ready to send your PostScript file from the DOS machine to the LaserWriter over the serial line using the XON/XOFF flow control protocol. Although DOS does not support XON/XOFF, the program COMXON.C given in Listing 2 can be used. Compile this program (I used Turbo C) to obtain the file COMXON.EXE. Then send your PostScript file (let's call it MYFILE.PS) with the following DOS command:

```
A>comxon myfile.ps
```

For users who don't own a PostScript printer, the question of how to use a remote laser printer arises.

You'll know the transmission has been completed when you see a report on the computer's screen that gives the number of XON's and XOFF's that have been sent. For a 12-page document printed on the LaserWriter Plus, there were about 15 of each, but when I used the new LaserWriter II-NT there were none (presumably because of the large buffer memory). One other tip: If you intend to print another document right away, send a Ctrl-D to the LaserWriter first. You can do this by creating a file CTRLD with

```
A>copy con ctrld
^D <enter>
^Z <enter>
```

Then use the command:

```
A>copy ctrld com1
```

to clear the LaserWriter for the next job.

Copying Files Across DOS/Macintosh Diskettes

If the LaserWriter is hooked up to a Macintosh, you'll have to find a way to transfer your DOS files to a Mac format diskette. You have several alternatives. One is to do this via two file transfer programs—one running on a Mac, one on a DOS machine—that support the same file transfer protocol (e.g., Kermit). A slicker (but more expensive) way to do the same thing is to buy a product called QuickShare. However, this requires that both a

Mac and DOS machine be available in the same room.

Another approach is to add a Mac disk-controller to your DOS machine (e.g., Matchmaker from Micro Solutions) and an external Macintosh $3^{1}/_{2}$-inch drive to your DOS system. This will let you make $3^{1}/_{2}$-inch Mac format diskettes. A new product form Central Point Software called the Copy II PC Deluxe Option Board can also be used for this purpose. It has the advantage that you can use a $3^{1}/_{2}$-inch drive installed for DOS to read and write Mac format diskettes.

A third alternative is to add a DOS disk-controller and $5^{1}/_{4}$-inch drive to the Mac. Apple sells such a product combination, but it can be used only on the Macintosh SE or the Macintosh II, since it needs an internal expansion slot. A more flexible product that uses the SCSI port found on all versions of the Mac is available from Dayna Communications.

Of course, all of these options run into money, so it may make more sense at this point to start thinking about putting that money into a PostScript printer of your own. The LaserWriter Plus did carry a list price of about $7,000, but it has recently been replaced by the LaserWriter II family. The new LaserWriter II-NT lists for around $4,600. There's a good chance that we'll soon see some PostScript clone printers at half that price. In the meantime, owners of PostScript printers already hooked up to DOS machines may be willing to offer a service for printing DOS-produced PostScript files, including services by mail (there's one about to start up in my area). And when you're ready to let the whole world see your desktop magnum opus, professional print shops will run your PostScript files through high-resolution (1200 or 2400 dpi) Linotronic printers, which produce their output on photographic paper with a quality that is acceptable to even the most fastidious of publishers. □

Did you find this article particularly useful?
Circle number 12 on the reader service card

Product Information

LaserWriter II-SC	$2,799
LaserWriter II-NT	$4,599
LaserWriter II-NTX	$6,599

Apple Computer Company
20525 Mariani Avenue
Cupertino, CA 95014
(408) 996-1010
Circle reader service #263

Quickshare	$465

Compatible Systems
P.O. Box 17220
Boulder, CO 80308
(800) 356-0283
Circle reader service #264

DaynaFile	$650 – $1,105

Dayna Communications
50 South Main Street, Suite 530
Salt Lake City, UT 84144
(800) 531-0600
Circle reader service #265

Matchmaker	$149

Micro Solutions
132 West Lincoln Highway
DeKalb, IL 60115
(815) 756-3411
Circle reader service #266

Copy II PC Deluxe Option Board	$159

Central Point Software
15220 NW Greenbrier Parkway
Beaverton, OR 97006
(503) 690-8090
Circle reader service #268.

Listing 2. C source code for an XON/XOFF file transfer utility

```
/* comxon.c  Sends file to com1 port using XON/XOFF flow ctrl */
#include "stdio.h"  /* NOTE: First use mode to set com params */
#include "dos.h"    /*  e.g.   A>mode com1:9600,n,8,1,p   */
#define XON 17      /* For most printers, connect com1 to printer */
#define XOFF 19     /* with null modem (modem eliminator) cable*/
#define COMPORT 0   /* BIOS uses 0 for com1: port,  1 for com2: */
int Xoncnt=0, Xoffcnt=0, Wait=0;

main(argc,argv)  /**************  START OF MAIN  ******************/
int argc;   char *argv[];
{
  int c;    FILE *in;
  if(argc!=2) {printf("Correct usage is:   comxon  filename\n"); exit();}
  if((in=fopen(argv[1],"rb")) == 0) {printf("Can't open %s\n",
          argv[1]);exit();}

  for(c=getc(in);c!=EOF;c=getc(in))
    {tstflow();          /* See if external device wants to talk to us */
     if(comwrite(c)!=0) printf("Transmission error for char %c\n",c);
    }
  fclose(in);
  fprintf(stderr,"There were %d XON's and %d XOFF's\n",Xoncnt,Xoffcnt);
}    /**********************  END OF MAIN  ***************************/

tstflow() /* Looks for char from external device; if XOFF wait for XON */
{
  int tstch;
  do{
     while(comstatus() != 0) /* while data arrives from external device*/
      { tstch=comread();          /* read arriving char */
        if(tstch==XOFF)           /* if it's XOFF */
          {Wait=1; Xoffcnt++;}   /* set Wait to true, count it */
        else if(tstch==XON)       /* But if it's XON */
             {Wait=0; Xoncnt++;} /* set Wait to false, count it */
      }
    }while(Wait == 1);   /* do loop while Wait signal is true */
}

comstatus()   /* BIOS service returns 0 if no data arriving at com1 */
{
  union REGS r;
  r.h.ah = 3;                           /* Status service of int 20 */
  r.x.dx = COMPORT;                     /* Select com port */
  int86(20, &r, &r);
  return( (int) r.h.ah & 0x01 ); /* Return bit 0 = data ready signal */
}

comread() /* BIOS service - reads 1 char from com1 input data line */
{ union REGS r;
  r.h.ah=2;
  r.x.dx=COMPORT;
  int86(20,&r,&r);
  return( (int) r.h.al);              /* Returns AL contents = char rcvd */
}

comwrite(c) /* BIOS service which sends 1 char out com1 xmit data line */
int c;
{ union REGS r;
  r.h.ah=1;
  r.x.dx=COMPORT;
  r.h.al=c;
  int86(20,&r,&r);
  return( (int) r.h.ah & 0x80);    /* Returns 0 if char sent out ok */
}
```

by Bob Blacher

Setting Up a Bulletin Board System

When we talk about bulletin boards, most people think of a public BBS that distributes shareware and has message systems available to the public. That certainly is one common use for BBS software, but by no means the only one. Many businesses have adopted BBS software for internal purposes. They have found that a limited-access BBS is an excellent way to exchange electronic mail and files among employees and clients. IBM, Apple, and Microsoft, for example, all have extensive, in-house bulletin board systems. Maybe your company should, too. This column will take a look at BBS software and what's involved in setting up such an in-house system.

BBS Software Choices

There's an almost bewildering array of BBS software available from which to choose. Which software is right for your company depends mostly on the intended use of the BBS, and the hardware you have available. Are you going to use the BBS mostly for messages or for file transfers? Who is going to have access to the BBS? Do you want all users to be able to see the same messages and files, or do you want different message bases and files for different categories of users? How many incoming lines do you need to handle the expected load? If you need more than one incoming line, what sort of budget do you have for the necessary hardware, and what level of network expertise is available in your organization? Do you have special requirements that can only be met if you have the source code for the BBS software?

I can't begin to cover all of the different packages that are available. Popular BBS software includes RBBS-PC, Fido, TBBS, DBBS, PC-HOST, Opus, WildCat, and many others. Some of the above packages are free (e.g., RBBS-PC, for which the QuickBasic source is also available). Other BBS packages are offered as shareware, and quite a few are now commercial products. (Several of these products were reviewed in *Micro/Systems Journal*, September/October 1987.)

I'm going to focus on a once-shareware, now-commercial BBS software package called PCBoard, for the simple reason that PCBoard is what I run on my system —it's the package I know best. I've also run RBBS-PC, Fido, and PC-HOST, but the basic considerations involved in setting up a BBS are almost the same.

How Hard Is It To Set Up a BBS?

It would take me about five minutes to get a PCBoard installation up and running in a minimal configuration. I'd copy the files to the hard disk, run the PCBSETUP program, and make a few choices about how the system is to be configured. That's all. It seems simple, but in reality the process is much more difficult.

After several years of running PCBoard, I know the software extremely well. A first-time user of the software, particularly one who has never set up a BBS had better set aside a lot longer than five minutes. Although I doubt anyone will need to read the hundreds of pages of documentation that come with PCBoard (does anyone really read all of the manuals that come with a piece of software?), some use of the manuals and some trial and error will be inevitable. Plan to set aside a day to get any BBS package up and . . . walking. Expect another week of work before your system stops generating error messages in the log file and crashing every now and then. Finally, plan on a month or more before you're really comfortable enough with the package to be able to run a highly reliable, well-tuned system that fully exploits the capabilities of the software.

What Hardware Do I Need?

The absolute, bare-bones configuration for any BBS software is as follows:

1. *A 100-percent compatible DOS system sporting at least a 10-MB hard disk.* My BBS has two hard drives, each with an 80-MB capacity, and a 40-MB Bernoulli attached to the machine for auxiliary storage and backup. Needless to say, the amount of storage space you need depends primarily on the number of files you're going to have available for downloading, but even a BBS that is primarily oriented towards electronic mail will need a hard disk, and the bigger the better.
2. *A "Hayes-compatible" modem operating at the top speed of your choice (and budget).* I think it's fair to say that 300-bits-per-second (bps) modems are now nearly obsolete and the same may soon be true for 1200-bps modems. 2400 bps is a comfortable speed for reading mail and transferring most files. If you can deal with the current incompatibility problems, 9600-bps modems make large file transfers a pleasure. In a corporate setting, it may be possible to standardize around one 9600-bps modem. If not, keep in mind that few 9600-bps modems from different manufacturers can communicate with each other at speeds higher than 2400 bps.

These are the minimum requirements to run a BBS with one incoming line. Going up to two incoming lines with PCBoard is fairly simple and inexpensive. In addition to the second modem, you'll need to upgrade the host computer to at least an 8-MHz, 80286-based machine for decent performance and purchase one of the popular multitasking packages. PCBoard sysops tend to choose DoubleDOS by SoftLogic Solutions, TaskView (recently enhanced and renamed OmniView) by Sunny Hill Software, or DESQview from QuarterDeck Office Systems. I've used both DoubleDOS and TaskView/OmniView to run two BBS nodes on an AT. (I'm

Robert Blacher operates the popular Computer Connections PCBoard bulletin board system in Washington, D.C. It is quite busy, but those willing to place their modems on redial may call (202) 547-2008. In his other life, Bob is an attorney.

currently using OmniView.) I've found DESQview too slow to handle dual communications tasks on an AT-class machine. On a 20-MHz 386 machine, DESQview, in combination with a memory manager like QuarterDeck's own QEMM or Qualitas' 386-MAX, becomes a much better environment. As I write this, Microsoft Windows is completely inadequate when running more than one node of any BBS software, even on a 386. Maybe someday there will be OS/2 BBS software that effectively allows multinode operation on one machine. It doesn't exist yet.

Going beyond two nodes with PCBoard requires a multiple-CPU system, which includes both slave cards (Alloy slave cards with NTNX are popular) or a true local area network system. PCBoard was written to be NetBIOS compatible. Any LAN software that follows DOS 3.x file sharing rules should work well with PCBoard including all of the "major" network systems such as Novell and 3Com. Depending on your requirements, a less expensive, peer-to-peer resource sharing system may be exactly right. A recent favorite among PCBoard sysops has been LANtastic by Artisoft. LANtastic does not require a dedicated server, is remarkably low-cost, allows resource sharing among machines, and provides adequate performance to run several PCBoard nodes. Needless to say, if you will be using up to 99 incoming lines (the theoretical maximum for PCBoard), you're looking at a substantial hardware and software investment, and you will need a LAN that delivers maximum performance.

Networking PCBoard, or any BBS software, means sharing the critical files. That means having just one file where all the information about your users is stored, shared message bases that are updated immediately, and shared access to files available for download. PCBoard allows this under the multitaskers mentioned above (the DOS SHARE program is run before the multitasker is loaded to provide "network" support) and under all NetBIOS-compatible systems. Other BBS software packages may use a network-specific file- and record-locking approach, and you'll need to investigate compatibility issues fully before choosing hardware and software.

The bottom line on hardware is the faster the better. BBS software, particularly when supporting multiple users, is very resource intensive. It may well be the most demanding task you've ever asked of your equipment. The faster the CPUs and the hard disks are, the better. If you've got the RAM to do it, use of RAM disks and disk caches can improve throughput enormously. It becomes mostly a question of what your budget can stand and what your users demand in terms of performance.

BBS Configuration Choices Add Flexibility

Part of the learning curve in installing BBS software is understanding the enormous flexibility that the software provides in how you configure your system. For example, PCBoard supports multiple message bases and file areas, referred to as "conferences." You can choose to allow only some of your users into certain conferences, thus restricting their access to only a subset of the messages and files on your system. A common configuration is to have the MAIN area open to all users, but to restrict access to conferences for the sake of confidentiality. Security concerns aren't the only issue, however. It may be that some of your users only care about a certain subject. Imagine a software company supporting both Apple and IBM users. An "Apple" conference and an "IBM" conference may be just the way to organize that BBS, and it can be made quite transparent to the callers—they can be automatically brought into the conference that matches their interests each time they call.

Security concerns will definitely gov-

CALL ADVERTISER DIRECTLY

ern certain decisions you're going to make when setting up the software. An in-house, corporate BBS may be configured to allow access only to those whose use has been authorized in advance. A company could use BBS software both for in-house electronic mail and for public access, such as client contact or product support. The flexibility is there. You'll just need to take some time to think through the type of access you wish to permit on your system and how it can be accomplished most sensibly with the BBS software you have chosen.

Some Examples of Corporate BBS Use

There are over 2300 PCBoard installations as I write this, and the authors of PCBoard estimate that their customer base is equally divided between public-access bulletin boards and corporate users.

Not surprisingly, members of the computer industry have been among the first to see the advantages of BBS systems, particularly for customer support. Some examples:

Iomega Corporation runs a multinode PCBoard system that is open to the public. As I happen to own a Bernoulli Box, I check in there from time to time for operating information and to download software updates. Iomega also has non-public systems that are used for dealer support and in-house communications.

When AST wanted to get LIM 4.0 drivers out to previous owners of their memory boards, making the file available on their multinode OPUS BBS system was an inexpensive and fast method of distribution. AST is now using another BBS software package to offer customer support and technical bulletins on their BBS.

QuarterDeck Systems runs a PC-HOST system as one means of product support to field customer inquiries and offer downloads of select files. The software offers sufficient flexibility so QuarterDeck can offer additional services to its "priority support" customers.

The above examples were chosen because they are systems I've used and because they happen to represent three different BBS software packages. (I'm trying to stay out of trouble here. People get very loyal to the BBS software they've chosen!) The range of corporate bulletin boards is very broad. To take another example with which I am familiar, a law firm could set up a BBS for a number of different purposes:

In-house use: A BBS is ideal to exchange mail and files (contracts, briefs, memoranda, etc.). This type of service can be particularly valuable for firms that operate in more than one location.

Client contact: Many firms mail newsletters or other informational, quasi-promotional materials to stay in contact with, and (ahem) encourage business from their clients. As clients have become increasingly sophisticated in communication via computer, a limited-access BBS system becomes a natural way to provide this service. File transfers via BBS may also be significantly less expensive and more timely than overnight courier or FAX. (In my experience, all lawyers do everything at the last possible moment.) A carefully designed BBS can be configured so that a client can upload a file that would not be available to anyone outside the firm. Similarly, a file can be made available for download by a single client only.

There's no reason why one BBS system couldn't support both of the above purposes. Of course, attorneys are very concerned about confidentiality. Properly configured, a BBS system can provide it. The column you are reading right now was first downloaded by an editor at *Micro/Systems* from a quasi-public BBS system of mine that serves several other purposes. Nobody else who called the system even knew the file existed, and nobody else could have downloaded that file, even if they had been aware of its presence and knew the file name. (Experienced PCBoard sysops will understand that the file was password-protected in FSEC with the caller's login password, thus making the security transparent even to the editor!)

Ready To Go?

Before you get started, give some consideration to the time required to maintain a BBS. Sysops of public BBS systems will tell you that the maintenance burden is substantial. That's true because they spend time registering and validating new users, responding to mail, and screening files posted for download.

The time commitment needed to maintain a non-public BBS can range from a few minutes a day to much, much more. Responding to customer queries on a BBS is probably no more time consuming than handling them by phone, and it's faster than handling them by mail. However, don't exclude maintenance time from your planning and remember to designate at least one system operator with sufficient time to become expert in the care and feeding of your BBS system.

After all, you want your BBS system to reflect the excellence of all of your company's work, right? □

Did you find this article particularily useful? Circle number 14 on the reader service card

Product Information

PCBoard

Clark Development Company, Inc.
P.O. Box 71365
Murray, UT 84107
(800) 356-1686 (New orders only)
(801) 261-1686 (Voice)
(801) 261-8974/5/6/7
(Data: USR HST modems)
(801) 261-8979
(Data: Hayes V-Series)

Circle reader service #263

Concurrent DOS 386

Think small in a big way

When you think multiuser/multitasking, think Concurrent™ DOS 386, the big name in small systems from Digital Research®, architects of the first standard operating system for personal computers. Now, Concurrent DOS 386 allows multiple users to share peripherals, files and applications, using serial terminal workstations linked by RS-232 cables to the system. It's fast, reliable and economical.

The big news today is small systems

Concurrent DOS 386 meets the increasing demands placed on small systems by supporting multiple DOS programs on both the system console and attached terminals. You can run popular programs such as Lotus® 1-2-3®, dBase® III, WordPerfect® and many more, with full math coprocessor support. The system runs up to 255 tasks simultaneously, with full intertask communications and byte-level record, file and device locking.

For people who hate waiting in line

Concurrent DOS 386 brings you all the remarkable speed and power of the Intel® 80386 processor. A prioritized preemptive scheduler allows task execution and intertask communication by several users at near full processor speed while letting some tasks "interrupt" others according to the needs of each user.

A small system with a big memory

Concurrent DOS 386 gives you access to four gigabytes of linear physical memory. Its powerful memory paging capability fully supports the Expanded Memory Specification with no additional hardware or software.

Menus at a touch

Now you can create and customize menus, while programmable function keys let you condense complex commands to a single keystroke. The file manager runs standard operating system functions, plus you have an on-line help facility, text editor and support for DOS-based device drivers.

Multiuser color graphics

Now with the introduction of the newest member of the Concurrent DOS family, Concurrent DOS 386/Multiuser Graphics Edition, your demands for high-resolution EGA bit-mapped graphics in the workstation environment can be met. Take advantage of advanced technology allowing you to run popular DOS-based graphics programs on individual workstations as well as on the system console without sacrificing system performance. Ask us about this exciting new version of Concurrent DOS 386.

All you have to remember is Concurrent DOS 386

Concurrent DOS 386 from Digital Research is the name to remember when it comes to 386 technology. The power and versatility of Concurrent DOS 386 are giving a new meaning to the word multiuser.

CALL DIGITAL RESEARCH AT 1-800-443-4200 AND ASK FOR OUR CONCURRENT DOS PROGRAMMER INFORMATION KIT.

CONCURRENT DOS 386:
SHARING THE SYSTEM AFFORDABLY

DIGITAL RESEARCH®

CIRCLE 98 ON READER SERVICE CARD

New Products

New Software Products

Extensible Editor Now Ready For OS/2

A new, fully extensible editor for OS/2 programmers has begun shipping from Lugaru Software, Ltd. The Epsilon Programmer's Editor offers a full C-like macro language, a completely redefinable keyboard, macro source code to all editor commands, and true concurrent compilations. The editor comes with an Emacs-like keyboard configuration.

Epsilon sells for $195. For more information, contact **Lugaru Software, Ltd.**, 5843 Forbes Avenue, Pittsburgh, PA 15217; (412) 421-5911.

Circle reader service #20

WordTech Ships dBXL for LAN

WordTech Systems has started shipping dBXL/LAN, the network version of the company's dBASE III+ superclone, dBXL Diamond Release. dBXL/LAN will run all existing dBASE applications without modification, and it is compatible with dBASE LAN Administrator to the extent that the two products can share the same files on the same network simultaneously. dBXL/LAN has also been upgraded with automatic refresh of user's screens when the data is modified, automatic record and file locking to allow single-user applications to be run in multiuser form, and the ability for multiple users to browse the same file. The network dBASE clone also features message exchanging, screen swapping, and distributed processing from remote workstations. dBXL/LAN also includes a copy of Networker Plus, which provides such features as distributed processing, screen and message transmissions, interactive communications, user identification, workstation control, and a flag system. The software will operate on networks supporting DOS 3.1 or later. Communications require NetBIOS. dBXL/LAN is also compatible with Quicksilver.

dBXL/LAN has an average cost of about $60 per user in a 10-user LAN. For more information on the product and pricing, contact **WordTech Systems, Inc.**, 21 Altrarinda Road, Orinda, CA 94563; (415) 254-0900.

Circle reader service #21

An Ounce of Prevention Against Disk Disaster

Prime Solutions Inc. has introduced Disk Technician Advanced, a new software package that is designed to repair and maintain DOS-based hard disk drives. Disk Technician Advanced automatically corrects both media- and data-related hard disk errors before they cause problems. Upon startup, the program first tests the hard disk and moves all data to safe areas on the disk. If it finds that a reliability problem exists, it automatically performs a low-level reformat and resets the interleaving to the optimal-tested "speed limit." Then it proceeds to routine testing, including defaults for operation from hard disk or floppy drive, and testing of all DOS partitions in one operation. Also included are tests for AI pattern recognition, four seek tests, an optional safe zone, an automatic screen saver, windowed help screens, and system, data, and File Allocation Table (FAT) recovery for downed systems. Disk Technician Advanced

C CODE FOR THE PC

source code, of course

	MS-DOS File Compatibility Package (create, read, & write MS-DOS file systems on non-MS-DOS computers)	$500
	Bluestreak Plus Communications (two ports, programmer's interface, terminal emulation)	$400
NEW!	Turbo Programmer (database application generator; source for library only; specify Turbo C or Microsoft)	$370
	CQL Query System (SQL retrievals plus windows)	$325
	GraphiC 4.1 (high-resolution, DISSPLA-style scientific plots in color & hardcopy)	$325
	Barcode Generator (specify Code 39 (alphanumeric), Interleaved 2 of 5 (numeric), or UPC)	$300
	Vmem/C (virtual memory manager; least-recently used pager; dynamic expansion of swap file)	$250
	PC Curses (Aspen, Software, System V compatible, extensive documentation)	$250
	Greenleaf Data Windows (windows, menus, data entry, interactive form design)	$220
	Vitamin C (MacWindows)	$200
	Greenleaf Communications Library (interrupt mode, modem control, XON-XOFF)	$175
	TurboTEX (TRIP certified; HP, PS, dot drivers; CM fonts; LaTEX)	$170
	Essential resident C (TSRify C programs, DOS shared libraries)	$165
	Greenleaf Functions (296 useful C functions, all DOS services)	$160
	Essential C Utility Library (400 useful C functions)	$160
	Essential Communications Library (C functions for RS-232-based communication systems)	$160
	WKS Library Version 2.0 (C program interface to Lotus 1-2-3, dBase, Supercalc 4, Quatro, & Clipper)	$155
	OS/88 (U••x-like operating system, many tools, cross-development from MS-DOS)	$150
	ME Version 2.0 (programmer's editor with C-like macro language by Magma Software; Version 1.31 still $75)	$140
	Turbo G Graphics Library (all popular adapters, hidden line removal)	$135
	CBTree (B+tree ISAM driver, multiple variable-length keys)	$115
	Minix Operating System (U••x-like operating system, includes manual)	$105
	PC/IP (CMU/MIT TCP/IP implementation for PCs)	$100
	B-Tree Library & ISAM Driver (file system utilities by Softfocus)	$100
	The Profiler (program execution profile tool)	$100
	Entelekon C Function Library (screen, graphics, keyboard, string, printer, etc.)	$100
	Entelekon Power Windows (menus, overlays, messages, alarms, file handling, etc.)	$100
	TurboGeometry (library of routines for computational geometry)	$90
	QC88 C compiler (ASM output, small model, no longs, floats or bit fields, 80+ function library)	$90
	Wendin Operating System Construction Kit or PCNX, PCVMS O/S Shells	$80
	C Windows Toolkit (pop-up, pull-down, spreadsheet, CGA/EGA/Hercules)	$80
	JATE Async Terminal Emulator (includes file transfer and menu subsystem)	$80
NEW!	Lisp-to-C Translator (includes Lisp interpreter, Prolog, and simple calculus prover)	$80
	MultiDOS Plus (DOS-based multitasking, intertask messaging, semaphores)	$80
	WKS Library Version 1.03 (C program interface to Lotus 1-2-3 program & files)	$80
NEW!	TE Editor Developer's Kit (full screen editor, undo command, multiple windows)	$75
	Professional C Windows (lean & mean window and keyboard handler)	$70
	lp (flexible printer driver; most popular printers supported)	$65
	Quincy (interactive C interpreter)	$60
	PTree (parse tree management)	$60
NEW!	MicroFirm Toolkit (28 Unixesque utilities for MS-DOS)	$50
NEW!	XT BIOS Kit (roll your own BIOS with this complete set of basic input/output functions for XTs)	$50
	HELP! (pop-up help system builder)	$50
	Multi-User BBS (chat, mail, menus, sysop displays; uses Galacticomm modem card)	$50
	Make (macros, all languages, built-in rules)	$50
	Vector-to-Raster Conversion (stroke letters & Tektronix 4010 codes to bitmaps)	$50
	Coder's Prolog (inference engine for use with C programs)	$45
	Virtual Memory System (least recently used swapping)	$40
	C-Notes (pop-up help for C programmers ... add your own notes)	$40
NEW!	Heap I/O (treat all or part of a disk file as heap storage)	$40
	Biggerstaff's System Tools (multi-tasking window manager kit)	$40
	PC-XINU (Comer's XINU operating system for PC)	$35
	CLIPS (rule-based expert system generator, Version 4.2)	$35
	Tiny Curses (Berkeley curses package)	$35
NEW!	SP (spelling checker with dictionary and maintenance tools)	$30
	Clisp (Lisp interpreter with extensive internals documentation)	$30
	Translate Rules to C (YACC-like function generator for rule-based systems)	$30
	6-Pack of Editors (six public domain editors for use, study & hacking)	$30
	Crunch Pack (14 file compression & expansion programs)	$30
NEW!	Pascal Compiler & Interpreter (P-codes, standard Pascal)	$25
	ICON (string and list processing language, Version 7)	$25
	FLEX (fast lexical analyzer generator; new, improved LEX)	$25
	LEX (lexical analyzer generator; an oldie but a goodie)	$25
	Bison & PREP (YACC workalike parser generator & attribute grammar preprocessor)	$25
	AutoTrace (program tracer and memory trasher catcher)	$25
NEW!	Data Handling Utilities in C (data entry, validation & display; specify Turbo C or Microsoft)	$25
	Arrays for C (macro package to ease handling of arrays)	$25
NEW!	ANSI Forms (forms manager based on ANSI codes)	$20
	C Compiler Torture Test (checks a C compiler against K & R)	$20
	Benchmark Package (C compiler, PC hardware, and Unix system)	$20
	TN3270 (remote login to IBM VM/CMS as a 3270 terminal on a 3274 controller)	$20
	A68 (68000 cross-assembler)	$20
	List-Pac (C functions for lists, stacks, and queues)	$20
	XLT Macro Processor (general purpose text translator)	$20
	C/reativity (Eliza-based notetaker)	$15

Data

WordCruncher (text retrieval & document analysis program)	$275
DNA Sequences (GenBank 52.0 including fast similarity search program)	$150
Protein Sequences (5,415 sequences, 1,302,966 residuals, with similarity search program)	$60
Dictionary Words (234,932 words in alphabetical order, no definitions)	$60
U. S. Cities (names & longitude/latitude of 32,000 U.S. cities and 6,000 state boundary points)	$35
The World Digitized (100,000 longitude/latitude of world country boundaries)	$30
KST Fonts (13,200 characters in 139 mixed fonts: specify TEX or bitmap format)	$30
USNO Floppy Almanac (high-precision moon, sun, planet & star positions)	$20
NBS Hershey Fonts (1,377 stroke characters in 14 fonts)	$15
U. S. Map (15,701 points of state boundaries)	$15

The Austin Code Works
11100 Leafwood Lane
Austin, Texas 78750-3409 USA

acw!info@uunet.uu.net

Voice: (512) 258-0785
BBS: (512) 258-8831
FAX: (512) 258-1342

Free surface shipping on prepaid orders For delivery in Texas add 7% MasterCard/VISA

CIRCLE 62 ON READER SERVICE CARD.

works at the controller/hardware level to detect single-bit/single-occurrence soft errors. It also keeps a record of failure patterns.

Disk Technician Advanced is available in non-copy-protected form for $189.95. For more information, contact **Prime Solutions Inc.** 1940 Garnet Avenue, San Diego, CA 92109; (619) 274-5000.

Circle reader service #22

Clipper Library Available for Novell TTS

The NetLib Network Library for Clipper developed by Communications Horizons now performs dynamic transaction rollbacks with Novell's Transaction Tracking System (TTS). TTS, which is available with NetWare Version 2.0 and above, takes "snap shots" of each record before that record is modified by a program. These records are then stored in an invisible transaction file until the transaction is successfully completed. Should that transaction abend, TTS rolls back the transaction so the original data replaces the modified records. With NetLib, users can access TTS, providing start, stop, and cancel commands. NetLib works with both the 86 and 87 versions of Clipper (libraries are included) and most versions of NetWare.

NetLib retails for $199. For more information, contact **Communications Horizons**, 701 Seventh Avenue, New York, NY 10036; (212) 724-0150.

Circle reader service #23

New Hardware Products

Winchester Disk Simulator For Drive Analysis

Flexstar Corporation has introduced the FS501 Winchester Simulator, a solid-state unit that characterizes control performance and calibrates disk test equipment for correlation of disk drive performance. The unit is self-calibrating and is not subject to mechanical or temperature-induced aberrations. It electronically simulates disk read/write functions as well as frequently encountered mechanical, electronic, environmental, and media-related faults. The FS501 standard for full parameter systems testing and analysis of all 3½- and 5¼-inch Winchester drives with ESDI (enhanced small drive interface) and ST506 interfaces. The unit can be programmed and used in a variety of ways to test results correlation between disk drives, controllers, and disk drive test equipment: as a standard, non-media drive for system operation confirmation, as a repeatable "worst case" drive for controller and system analysis; and as a portable test unit for field use. Eight levels of accurate, repeatable bit shift can be induced in the FS501 for calibrating test equipment or characterizing controller PLL (phase-lock loop) designs.

The FS501 Winchester Simulator is available for $2,995. For more information, contact **FlexStar Corporation**, 606 Valley Way, Milpitas, CA 95035; (408) 946-0407.

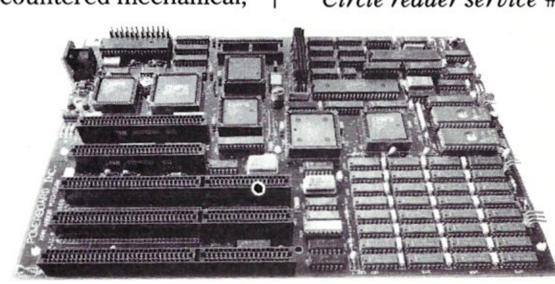
Circle reader service #26

New 286 Motherboard Boosts PC Power

Powersystem II from Powerboard, Inc., is a motherboard replacement that can increase system speed from four to 10 times. According to Powerboard sources, the new board gives IBM PC and XT users a cost-effective way to upgrade their PC to the 80286 with the performance level and features of the PS/2 Model 60. The motherboard replacement is completely compatible with all PC software, peripherals, and add-on boards, and comes with a full megabyte of on-board memory. Boards can be customized to meet individual needs. The board comes standard with a 10-MHz CPU, 1 MB memory, 8 and 16 bit bandwidth, and diagnostics in ROM. The software utility package supports 3½-inch floppies and includes RAM disk, auto RAM disk backup, EMS 4.0, and a ROM-embedded cache.

The volume prices for the Powersystem II is $995, including coprocessor. For more information, contact **Powerboard, Inc.**, 3965 Research Park Drive, Ann Arbor, MI 48102; (313) 668-1950.

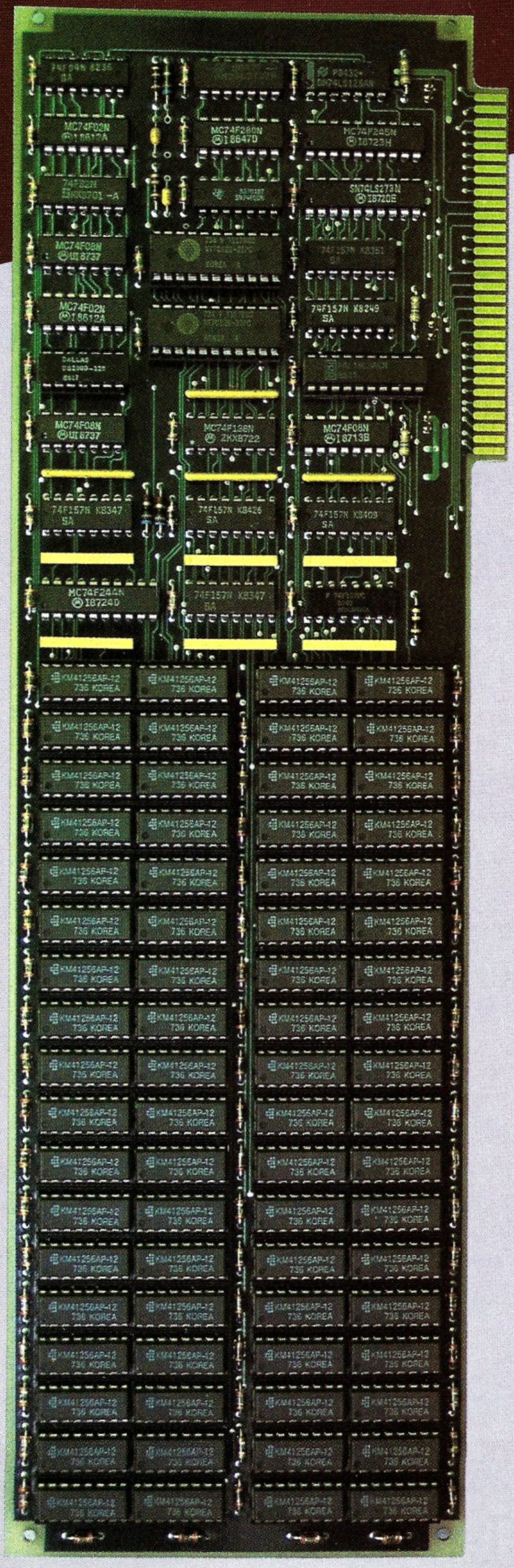
Circle reader service #27



Powersystem II

X-BANDIT

Designed for the EMS 4.0 Standard



DESIGN PHILOSOPHY

- The Teletek X-Bandit was specifically designed to utilize the advanced features of the Lotus/Intel/Microsoft EMS 4.0 Specification. It is available in both 8 and 16 bit versions for use in the IBM XT, AT, and compatibles.

MEMORY

- Segmented Memory Mapping allows the user to fill out unused memory segments between 640K and 1 Megabyte.
- Split Memory Addressing allows the user to fill out conventional memory to 640K.
- Extended Memory Addressing is available for the PC/AT version.
- 2 MB capacity in a single slot. Up to 8 MB per system.
- Parity checking.

SOFTWARE

- Easy menu-driven auto configuration software.
- Device driver includes print spooler and RAM drive.
- Supports multitasking with the appropriate shell-resident software package.

SPEED

- 6/8/10/12 MHz speed with 0 wait states. 16 MHz speed with 1 wait state.

WARRANTY

- One year parts and labor.

TELETEK

4600 Pell Drive
Sacramento, CA 95838
(916) 920-4600
Telex #499-1834

DOS system running Lotus 1-2-3

SCO XENIX system running SCO Professional

THIS IS AN IBM PS/2 MODEL 80 RUNNING DOS

Under DOS, this PS/2™ is a powerful 80386-based single-tasking, single-user computer that can run thousands of DOS applications. In 16-bit, 8086 mode.

One at a time.

When OS/2™ software becomes available, the PS/2 can become a multitasking, single-user computer running in 16-bit, 286 mode that can also single-task those DOS applications under OS/2.

One at a time.

With DOS or OS/2, the PS/2 will support one user.

	1 user (DOS)	1 user (OS/2)
Cost per system**:	$12,389	$12,594
Cost per user:	**$12,389**	**$12,594**

THIS IS AN IBM PS/2 MODEL 80 RUNNING SCO XENIX

Under SCO XENIX,® this PS/2 becomes a powerful 80386-based multitasking, multiuser computer that can run thousands of XENIX applications. In full-tilt, 32-bit, 386 mode.

Many at a time.

And using SCO VP/ix,™* the PS/2 can multitask DOS applications under SCO XENIX.

Many at a time.

With SCO XENIX, the PS/2 will support one user. Or 9 users. Or even 33 users.

And it can do all that today because you can get SCO XENIX for the PS/2—*now!*

	1 user	9 users	33 users
Cost per system**:	$14,559	$19,726	$40,402
Cost per user:	**$14,559**	**$2,192**	**$1,224**

SCO XENIX System V and the SCO XENIX family of software solutions are available for all industry-standard 8086-, 80286-, and 80386-based computers, and the IBM® Personal System/2™ Models 50, 60, and 80.

SCO™
THE SANTA CRUZ OPERATION

(800) 626-UNIX (626-8649)
(408) 425-7222
FAX: (408) 458-4227
TWX: 910-598-4510 SCO SACZ
uucp: ...decvax!microsoft!sco!info!

*SCO VP/ix available as separate product.

Cost comparisons are based on most recently published U.S. domestic suggested list prices. Cost model: Base machine: IBM PS/2 Model 80, 70Mb disk, 1Mb RAM, IBM 8512 color monitor, 1Mb additional IBM RAM, IBM ProPrinter XL,® **1-user DOS system: Base machine, plus DOS 3.3, WordPerfect 4.2, Lotus® 1-2-3,® dBASE III PLUS.® **1-user OS/2 system:** 1-user DOS system; substitute OS/2 for DOS. **1-user SCO XENIX system:** Base machine, plus SCO XENIX 386 for PS/2, SCO VP/ix, SCO Lyrix® (word processing), SCO FoxBASE+™ (dBASE III PLUS workalike), SCO Professional™ (1-2-3 workalike). **9-user SCO XENIX system:** 1-user SCO XENIX system, plus intelligent 8-user multiport card, 8 IBM 3151 ASCII terminals. **33-user SCO XENIX system:** 9-user SCO XENIX System, plus 3 more intelligent 8-user multiport cards, 24 more IBM 3151 ASCII terminals, 4 Mb additional RAM, additional 70Mb disk.

CIRCLE 94 ON READER SERVICE CARD