The Elusive Corporate Dial-up Networking Solution

BY JOHN E. JOHNSTON

This month's column promises to be another humbling experience for me as I must tell you how my department has floundered, for several years, in providing a dial-up networking solution for our end users. Let me give you a bit of history concerning our dial-up networking needs and the solutions that we have implemented to meet those needs.

MAINFRAME DIAL-UP ACCESS

When my current employer first hired me, I was responsible for mainframe support. We had a small (five-node) LAN, but at that time, the LAN was just an experiment. One of my first tasks was to provide dial-up access for the systems and applications programmers to allow them to correct problems that occurred on second and third shift. The solution that I implemented was simple, inexpensive, and very reliable. A 3708 protocol converter was attached to the mainframe with multiple 9600 baud modems. The programmers were given 3151 dumb terminals with external 9600 baud modems. This solution worked great. The response time was reasonable, and we rarely experienced problems with the setup.

OUR FIRST LAN DIAL-UP SOLUTION: THE NAS

As time progressed and the experimental LAN grew into a large production monster, I was given responsibility for it. My first task as LAN manager was to provide remote access for the LAN administrators to allow them to correct problems that occurred at night. Since all of our LAN software was purchased from Novell, I naturally sought a Novell solution. Novell's dial-up networking offering at that time was the NetWare Access Server (NAS).

The NAS creates virtual DOS sessions for each dial-up user, sharing the CPU between multiple users (much in the same way WinFrame operates). The NAS served our needs for about a year, by which time we had outgrown the capacity of the server.

Our network has grown substantially over the last year, and our users need access to applications and data residing on several platforms, including the mainframe and the NetWare and Windows NT networks.

NETWARE CONNECT

Our LAN continued to grow as more and more users were given PCs to replace their mainframe terminals. Many of our users needed modems to dial out to vendor-sponsored BBS systems and to CompuServe. Rather than placing modems and dedicated telephone lines in these users' offices, I was asked to provide a dial-out modem solution. Again, being a NetWare shop, I searched for a Novell solution and found NetWare Connect version 1.0. NetWare Connect required a large TSR to be loaded on the PC to re-direct the communications from the PC COM port to the NetWare Connect file server. This TSR created havoc as it pushed many of our DOS/Windows PCs into the RAM cram problem.

NetWare Connect 1.0 was buggy. These bugs, combined with the RAM cram problem caused by the dial-out TSR, made the NetWare Connect solution more trouble than it was worth. (NetWare Connect version 2 has made great strides and is now a viable remote access solution.) In the meantime, my credibility was taking a beating. It seemed that each solution I implemented was less stable than the previous solution.

IBM 8235

By now my boss was getting a bit perturbed with the existing LAN dial-up access and decided to give the mainframe guys a shot at providing access. While my ego was a bit bruised, I quickly let go of the responsibility and moved on to other projects. The mainframe guys implemented the IBM 8235. This box provided remote node access to the LAN. The 8235 was stable, but our users now needed to run large LAN-based applications remotely. The 8235, being a remote node solution, could not live up to these needs. Since loading a 2MB application would take several minutes, the end users quickly abandoned the 8235 and began screaming for a faster solution.

REMOTE CONTROL

The mainframe guys gladly turned over the remote LAN access project to my group again. The only viable alternative appeared to be some type of remote control solution. We looked into Citrix's WinFrame, but at the time, WinFrame was in its infancy and appeared to be just as unreliable as our past undertakings. We then purchased the EVERSYS

Corporation's CAPserver. The CAPserver is a large chassis into which multiple cards, each containing an Intel processor, are inserted. A digi-board also resides in the CAPserver and connects multiple modems to the box.

Remote users could dial into the CAPserver and then take control of one of the CPU cards in the box. After logging into the network, the users would start Windows and be on their way. This solution worked fine until we tried to implement some of our RAM-intensive DOS applications. The drivers, TSRs, and expanded memory requirements of these applications chewed up every bit of conventional memory these DOS-based CPU cards had, causing crashes and extreme unreliability.

WINFRAME

Now it's D-Day. My boss is totally disgusted with the state of our LAN remote access solution and has given me one last chance to fix the problem. (She didn't say what would happen if I failed again, but I try not to think about that!) Our network has grown substantially over the last year, and our users need access to applications and data residing on several platforms, including the mainframe and the NetWare and Windows NT networks. WinFrame is my last hope. I have installed a trial version of the system and am pleasantly surprised to find that almost all of our applications are working flawlessly. WinFrame, being a remote control solution, also provides very good response times from our applications. For more information on WinFrame, please refer to Guy Yost's article in the February 1998 issue of *Technical Support* magazine.

CONCLUSION

We are still in the early implementation phase of our WinFrame project, but it looks like it will be the silver bullet that I have been searching for. We are currently providing access to the following applications via the WinFrame:

◆ mainframe applications via the RUMBA emulation

◆ access to several UNIX boxes using the RUMBA emulator

◆ Microsoft Office Suite

◆ Microsoft Exchange (using the Outlook client)

◆ access to data residing on our Windows NT domains

◆ access to data and applications residing on our NetWare file servers

If you would like to see a full-length article on the technical details of our WinFrame implementation, please email me at johnj@fast.net or the editor at editor@naspa.net. **ts**

NaSPA member John E. Johnston is manager of technical support and communications for a major hospital in Pennsylvania. He designs and maintains cross–platform local and wide area networks utilizing NetWare, OS/2, DOS, and Windows.