Two MUMPS Networking Models

by Dayna Aronson and Richard Landis

To compete in today's multiplatform, multinetwork environments, software developers must ensure that their applications can be accessed across multiple networks. One health-care applications developer, Dendrite Clinical Systems, Inc. (Portland, Oregon), has successfully employed MUMPS in UNIX and Novell networking environments.

Dr. U. Scott Page, Dendrite's original program designer, said that his firm selected MUMPS for the patient analysis and tracking system (PATS) after some frustrating trial-and-error development experiences.

"We started out using BASIC, but it was too slow. Then we tried a relational database; however, the software couldn't handle the volume or the sophisticated analysis we needed," Page said. "Finally we settled on MUMPS and never looked back."

According to Page, MUMPS is extremely flexible and efficient in handling the large volumes of data generated by their multicenter (seventeen currently) databases. "It allows for variable-length records and is string-oriented, so we can add follow-up data at any time without jeopardizing the integrity of previous data."

Originally developed for cardiothoracic surgery practices, PATS tracks procedures and patients over time. Many current PATS users have extensive databases reaching back twenty years and containing more than ten thousand cases.

Originally developed for cardiothoracic surgery practices, PATS tracks procedures and patients over time. Many current PATS users have extensive databases reaching back twenty years and containing more than ten thousand cases.

Cardiothoracic surgeons, cardiac catheterization labs, ophthalmologists, cancer centers, organ transplant centers, and quality assurance departments are among PATS users today. Each database is connected to a central, patient-demographic file, which expands to include user-defined questions.

MUMPS, UNIX, and Novell

Dendrite's first experience linking UNIX and Novell was at Swedish Hospital in Seattle, Washington. Swedish uses PATS running under UNIX, with TCP/IP (Transport Control Protocol/Internet Protocol) to provide internetworking connectivity between the UNIX host and multiple local area networks (LANs) running Novell NetWare. TCP/IP was developed as a network protocol in the 1980s by the Defense Advanced Research Projects Agency (DARPA) for the U.S. Government's ARPANET wide-area network project.

Though often viewed as a temporary solution until new standards such as OSI (open systems interconnection) are implemented, TCP/IP is a tried-and-true standard supporting thousands of computer stations on local and wide-area networks. With more than a decade of use, TCP/IP has proven to be highly reliable and versatile.

Another reason for its popularity is that TCP/IP is in the public domain. As a result, developers and integrators can create implementations of TCP/IP without paying for licenses or expensive copyright fees. Additionally, the large number of established TCP/IP users represents a significant market that will continue to foster commercial development for the foreseeable future.

To understand how TCP/IP functions we need to look at the OSI Reference Model, developed by the International Standards Organization (ISO). The OSI reference model is a logical architecture establishing procedures for the exchange of information between computer systems.

ISO working groups defined seven specific layers for the reference model for open systems interconnection. They begin at the application layer, wherein users interact with a system, and progress to the physical layer, wherein data are converted into signals that are transported through physical media such as telephone wire.

Continued on page 56.

Application
Presentation
Session
Transport
Network
Data link
Physical

Figure 1. OSI reference model.

In the OSI reference model, the internet protocol (IP) works at the network layer and is a method for linking subnetworks. The network layer has the responsibility of transferring data between two systems and accomplishes this by means of an addressing scheme. By establishing unique addresses for all nodes (workstations) on interconnected systems, the network layer can relay data across intermediate systems, thereby providing connectivity between nonadjoining systems.

IP addresses are divided into four decimal parts, for example 255.22.33.6. The beginning one or two segments of the address identify a network. The ending segment(s), called the "host" portion, identifies a specific node. Each node on the internetwork must have a unique address.

When the IP receives a data packet from a node, it consults a routing table with the information needed to forward the packet to the proper destination. If the packet's destination is a node on a different network, one node serves as a router, with the job of consulting the routing tables for each network and forwarding the packet on to the other network.

The IP delivers data packets node-to-node and is classified as a connectionless service. In other words, it is an attempt to get data from node A to node B without the two nodes actively communicating to ensure error-free delivery.

TCP, on the other hand, is a stream- or connection-oriented service provider. TCP establishes a connection between two nodes or processes and maintains that connection until a successful data transfer has occurred. In the OSI reference model, TCP functions at the transport layer, one step higher than the network layer where the IP functions. TCP ensures that data arrive error-free and in the proper order.

Two Real-World Implementations

If UNIX is the platform of choice to run the MUMPS application, internetworking connectivity via TCP/IP is built-in because most commercial releases of UNIX have TCP/IP protocols included.

In the following implementation, Dendrite used SCO UNIX version 3.2.2 bundled with TCP/IP as the host operating system. InterSystems, Inc.'s M/SQL was the MUMPS kernel providing the operating environment. Novell's LAN "Workplace for DOS" was used for TCP/IP-based terminal emulation.

Swedish Hospital has a fiber optic network backbone with token-ring topology. Sixteen file servers run Novell NetWare 3.11 and one IBM 3090 mainframe. The installation required that all existing LAN workstations have access to the Dendrite PATS application and data. Using the open datalink interface (ODI) supplied with NetWare 3.11, Health Data Research, Inc., ran multiple protocols from the same work station.

The first step was to install the terminal emulation software needed to access the UNIX host. (Telnet is the generic name for TCP/IP-based terminal emulation applications.) LAN "Workplace for DOS" has a Telnet application called TNVT220 which provides DEC VT (Digital Equipment Corporation) and ANSI terminal emulation at each workstation. For this implementation, Dendrite installed LAN "Workplace for DOS" on the workstation's local hard drive.

Fortunately, some MUMPS developers have added networking utilities to their products to solve the problems.

Creating the IP address of the UNIX host and each workstation, allowing them to locate each other, came next. When installed locally, LAN "Workplace for DOS" created a subdirectory called TCP. In this directory, the installer created a file called "hosts," where LAN "Workplace for DOS" looked to identify valid IP addresses of nodes with which to communicate.

The first workstation file contained the following two lines:

```
89.0.0.2 Reg_2
89.0.0.1 tr
```

89.0.0.2 is the IP address of this particular workstation, for use when the Telnet application is active. 89.0.0.1 is the IP

Continued on page 58.

address assigned to the UNIX host. Address selection is arbitrary, providing that each address on the network is unique. The UNIX host must have a file containing its IP address. This file, also called "hosts," resides in the "/etc" subdirectory on the UNIX system. The name following the address is a convenient alias and may be used by many network applications in place of the actual IP address number.

The third requirement was getting the network interface cards in the workstations to handle the TCP/IP protocol stack as well as Novell's IPXODI (Internet Packet Exchange Open Datalink Interface) protocol. At this point, all that was needed was to add the line "TCPIP" to the original batch file.

Finally, the TCP/IP protocol driver, TCPIP.COM, supplied with "LAN Workplace for DOS," was placed in the same directory as the other programs executed from the batch file.

With all the parts in place, issuing the command "tnvt220 tr" from the DOS prompt on the workstation placed the user at the UNIX log-on prompt. Users then were able to log on and begin running their M applications.

MUMPS, DOS, and Novell

Many organizations are beginning to view operating systems such as DOS-based Novell NetWare as complete system platforms because of their ever-improving functionality. Consequently, integrating a MUMPS application into a Novell NetWare or similar environment rapidly is becoming a necessity. While the previously described UNIX-to-Novell integration is one way to meet this need, organizations that don't use UNIX need another option.

The problem is that many of the most popular implementations of M do not function as standard DOS-based network applications. Fortunately, some MUMPS developers have added networking utilities to their products to solve the problems.

Dendrite's second implementation uses proprietary network protocols to interface M applications directly with Novell networks. Dendrite's corporate offices use Micronetics Design Corporation's MSM-PC/386 as the MUMPS operating system, with MSM-NET and MSM-LAT providing protocol support for a Novell network with Ethernet topology.

Because M stores data on disk differently from Novell or 3COM, when more than one user attempts to access the M database file corruption may occur and disk input/output may be slowed significantly. The solution is to dedicate one PC on the network as a MUMPS server. This PC has a network interface card just as all other workstations on the Network. In addition, it has a hard drive that contains the M operating system, the M database, MSM-NET, and MSM-LAT. MSM-NET and MSM-LAT are utilities developed by Micronetics to facilitate network installations.

Using the MSM SYSGEN utility, Dendrite installation staff configured MSM-PC/386 with MSM-NET and MSM-LAT. Local area transport (LAT) is a protocol developed by Digital Equipment Corporation allowing multiple serial ports to be replaced by a single Ethernet card in a network.

In spite of the continuing talk about emerging new standards, MUMPS developers, with a little ingenuity and currently available tools, have no trouble meeting the integration requirements of most organizations.

Part of this process included the creation of a LAT service name. Workstations use this name to connect to the M server and access the application. Once the M server was properly configured, workstations needing access to the server were configured.

The first step was to replace the Novell internet packet exchange (IPX) protocol packet driver at the workstation with the IPXPDI.COM packet driver provided by Micronetics. This driver formats Ethernet data packets according to the LAT protocol. Next, terminal emulation software which supports LAT, such as Smarterm from Persoft, Inc., was installed on the workstation.

After starting the LAT software, Dendrite could now run its M application from LOGIN by typing the LAT service name of the application routine created with the SYSGEN utility on the MUMPS server.

While this article covers two examples of integrating M applications into network environments, there are other ways to accomplish the same purpose. It is important to note that Dendrite's implementations used well-established products and methods. In spite of the continuing talk about emerging new standards, M developers, with a little ingenuity and currently available tools, have no trouble meeting the integration requirements of most organizations. ❖

Richard Landis is the executive director and Dayna Aronson is a technical specialist at Health Data Research, Inc. (HDR). HDR is a nonprofit corporation that develops, uses, and analyzes computerized medical-outcomes databases. For more information, contact the authors at HDR, Suite 216, 5201 S.W. Westgate Drive, Portland, OR 97221.