Journal of Medical Systems, Vol. 10, No. 4, 1986

Local Network and Distributed Processing Issues in The Johns Hopkins Hospital*

Stephen G. Tolchin and Wendy Barta

The Johns Hopkins Hospital has initiated an ambitious program to apply modern technologies to the development of a new, comprehensive clinical information system. One component of this system is a networking technology for supporting the integration of diverse and functionally distinct information systems. This paper discusses the selection of the networking technology implemented at JHH, issues and problems, networking concepts, protocols, and reliability.

INTRODUCTION

The Johns Hopkins Hospital is a 1,000-bed teaching and treatment facility located in Baltimore, Maryland. The hospital currently has a collection of clinical information system components that have been developed and operated by different departments under a decentralized management structure. A preplanned approach for the efficient sharing and transfer of data did not exist when these systems were implemented. Therefore, ad hoc special-purpose, low-speed interfaces were developed among systems when a priority need existed. Generally, duplicative data entry into the various systems occurred as a result of the absence of comprehensive application-level integration. This is costly, and it has resulted in inconsistencies of the data in different systems. Furthermore, the automated support of clinical functions for the outpatient, for the inpatient, and for some of the ancillary services was minimal. Therefore, the JHH initiated development of a comprehensive clinical information system using modern technologies based on a strategic architectural plan. A key element of this plan is to achieve functional integration of the current and developing clinical systems by applying local network technology. The Operational and Clinical Systems (OCS) Division was formed to prepare and implement the plan. Considerable progress has been made in the application of networking technology. This paper describes issues, decisions, and implementations relating to the networking layer of the architecture.

From The Johns Hopkins Hospital, Baltimore, Maryland 21205, and the Applied Physics Laboratory.

* This paper was presented at the 19th Hawaii International Conference on System Sciences, Honolulu, Hawaii, January 1986, and is republished here with permission.

REVIEW OF CURRENT SYSTEMS AND ARCHITECTURE

The technical challenge of networking at JHH results in part from the diverse collection of existing hardware and software. The existing computer centers include the following:

1. An IBM mainframe shop consists of a 3081 and 3083 running VM/MVS in a CICS TP monitor environment. Clinical applications include an admissions, discharge, a transfer system with many add-on functions, and an inpatient pharmacy system, both developed using modified PCS/ADS tools. The hospital's financial systems run on these computers also. A separate patient identification system containing about 1.5 million records in a VSAM file runs on this computer; however, most outpatient services do not have on-line access to this system, nor are other clinical systems functionally integrated with this system. Information is obtained by telephone calls to Medical Records personnel who have on-line access.

2. A Department of Laboratory Medicine (DLM) Information System, which runs on three PDP 11/70 computers running InterSystems MUMPS M11 + native, can be reconfigured to operate with only two computers working. Low-speed interfaces connect this system with the IBM mainframe to pass admit and change information to DLM and to pass lab results back to the mainframe for (limited) display on 3278 terminals on the nursing units.

3. An Oncology Center system, which runs on two PDP 11/70 computers running InterSystems MUMPS M11+ and TEDIUM, is a sophisticated system that supports many clinical function needs of the Oncology Center. This system also runs a separate pharmacy system. It is connected by low-speed lines to the DLM for lab results transfer.

4. An Anesthesiology and Operating Room scheduling system runs on a PDP 11/84 under InterSystems MUMPS and TEDIUM. This system is stand-alone.

5. A VAX 11/750 computer runs VMS with InterSystems M/VX MUMPS, the Wollongong Eunice UNIX emulation system for VMS and the Relational Technology, Inc., INGRES relational data base management system. MUMPS is used to support the current Emergency Medicine system, which was developed using TEDIUM. A new system is being developed under the UNIX operating system using INGRES. The Emergency Medicine system is currently stand-alone, yet is keeps the only on-line clinical patient history in the institution.

6. The Johns Hopkins School of Medicine operates two PDP 11/70 computers running InterSystems MUMPS M11 + native. These computers provide professional fee billing services and scheduling services for the Johns Hopkins Internal Medicine Associates. These systems are stand-alone.

7. The Wilmer Eye Institute has recently acquired a VAX 11/750 running a variation of BSD 4.2 UNIX* and the INGRES data base management system. This system will be used primarily for research.

8. The Operational and Clinical Systems Division has acquired three Pyramid 98x superminicomputers running the OSx dual port of AT&T System V UNIX and BSD 4.2

* UNIX is a trademark of AT&T Bell Laboratories; PDP and VAX are trademarks of Digital Equipment Corp.; INGRES is a trademark of Relational Technology Inc.; Ethernet, XNS, Courier, and SPP are trademarks of Xerox Corp.; OSx and 98x are trademarks of Pyramid technology Corp.

UNIX. These systems are being used for many new development projects. The INGRES data base management system and both the DoD TCP/IP and Xerox XNS networking protocols run on these machines.

9. The Department of Radiology has several special touch-screen reporting stations that compose text reports and uplink these to the IBM mainframe. This enables radiology reports to be available on line immediately after readings in most cases. Plans call for changing the type of reporting stations and connecting these to the Pyramid minicomputer supporting the radiology information system under development. Reports would then be transferred across the network to workstations on clinical units as well as to the mainframe (for an interim period, to allow access from 3278s currently on the clinical units).

10. Several workstations are in use for the development of a workstation-based inpatient clinical system. These include Xerox Star and 6085 equipment running the XDE/MESA and Viewpoint environments and Sun Microsystems equipment running UNIX and AT&T UNIX PCs.

11. There also exist several departmental collections of MS-DOS PCs of various types along with some several PC networks.

Other computer systems are operating at the JHH for specialized purposes. None of these systems can process transactions with the IBM mainframe system to obtain PID information; consequently, there are several, mutually inconsistent, PID files extant. Similarly, functional interfaces to support a wide range of clinical and administrative support needs do not exist.

In addition to the computers described above, the OCS Division is developing several new systems, including the following:

12. A central data base server will be accessible from computers and terminals at high speeds across the network. This system is implemented on a general-purpose superminicomputer using the INGRES relational data base system. It contains records on approximately 2.1 million patients, including patient identification and demographics, clinical encounter/history summary, and some financial information.

13. A new Radiology Department information system will be integrated into the network for scheduling, film tracking, resource management, and other functions. A network of personal computers (MS-DOS-based) connected by a 3Com (Thin-net) Ethernet that serves the Department will be integrated with the system.

14. A new Emergency Medicine system will exist for complete on-line support, with special focus on clinical needs and urgency of care as well as administrative and financial functions.

15. A new inpatient system will itself be a distributed subsystem using workstation technology on nursing units. The workstations will be networked into the general net and will be closely coupled with a superminicomputer.

16. An outpatient clinical information system will support many of the outpatient clinics at JHH. Initially, registration, patient appointing to all areas and ancillary services, and resource scheduling will be supported, but the major long-term focus is clinical support.

All the above new systems are being developed on superminicomputers running the UNIX operating system and using the INGRES data base management system.

In addition to all these systems, there are many personal computers used for various functions. These devices pose some interesting problems: Sometimes they need to look like terminals to one or more of the larger systems, and sometimes they need to look like computers serving as producers or consumers of clinical information. We have developed the following policies regarding these devices: (a) They will not be connected electrically to the main hospital network to prevent accidental disruption of the network operations and to permit better control and management of the net. (b) Communities of PCs which are networked together and which need to access clinical system resources will be bridged or gatewayed into the net where possible. Since the network communication servers for the JHH Ethernet are from Bridge Communications, the 3Com Etherseries products provide a recommended way to network these PCs and to internet them with the JHH network. This is because of mutual compatibilities of the protocols for these products and the functionality of the 3Com network. (c) Individual PCs that need to access clinical system resources and do not do so by direct or modem connection to one or more machines will be supported where appropriate by 9,600 baud (or 1,200 baud) direct or dial-up connection to a network communications server.

The Johns Hopkins School of Medicine has also installed an Ethernet, and the two networks are connected by a gateway server to form an extended internet. Many other computers exist on this JHSM network including the system used by the Welch Medical Library for on-line bibliographic searches.

COMMUNICATIONS ARCHITECTURE REQUIREMENTS

In order to address the problems of decentralization of computing functions, it is necessary to provide a general-purpose, extensible, standardized communications mechanism among computers and user devices such as terminals and workstations. This mechanism is the lowest layer foundation of an architecture. The technology of local area communications networking can provide this foundation. The approach must be general enough to permit internetworking of several geographically remote medical centers to permit future exchange of clinical information. It is also desirable to internet easily with commercial and other packet data networks (PDNs) offering value added services such as medical data bases. Thus, any terminal in the hospital should be able to access such remote resources transparently by logical name without special program development or special hardware.

Ethernet

The network must be capable of interfacing to a wide variety of existing hardware and be a sufficiently recognized standard that future hardware offerings will be very likely to support the standard interface. A prime requirement is that the network be a nonproprietary industry standard. The only network offering that satisfies these criteria is the Ethernet technology. Ethernet, while originally a Xerox Corporation invention, is not a product. It is a technology that Xerox has made public domain and that has been standardized by the IEEE 802.3 local network standards committee. One constructs an Ethernet in a Tinker-Toy manner, selecting components from various vendors. Furthermore, although a standard exists, there are actually several versions of Ethernet: version 1, version 2 (adopted by a consortium consisting of Xerox, DEC, and Intel), and the

IEEE 802.3 version. Some incompatibilities naturally exist, and these must be understood by the design and installation team.

In order to satisfy the needs of remote access by a program executing in one computer to resources such as programs or data bases resident on another computer in an efficient manner, high-speed communications are needed. First, the medium itself must support a high band width with shared access from many devices, and second, inexpensive hardware interfaces to the medium must exist for many different types of computer. The Ethernet supports 10-megabit-per-second data rates on a shared coaxial cable medium, which is logically a broadcast bus and physically a limited-branching, limited-

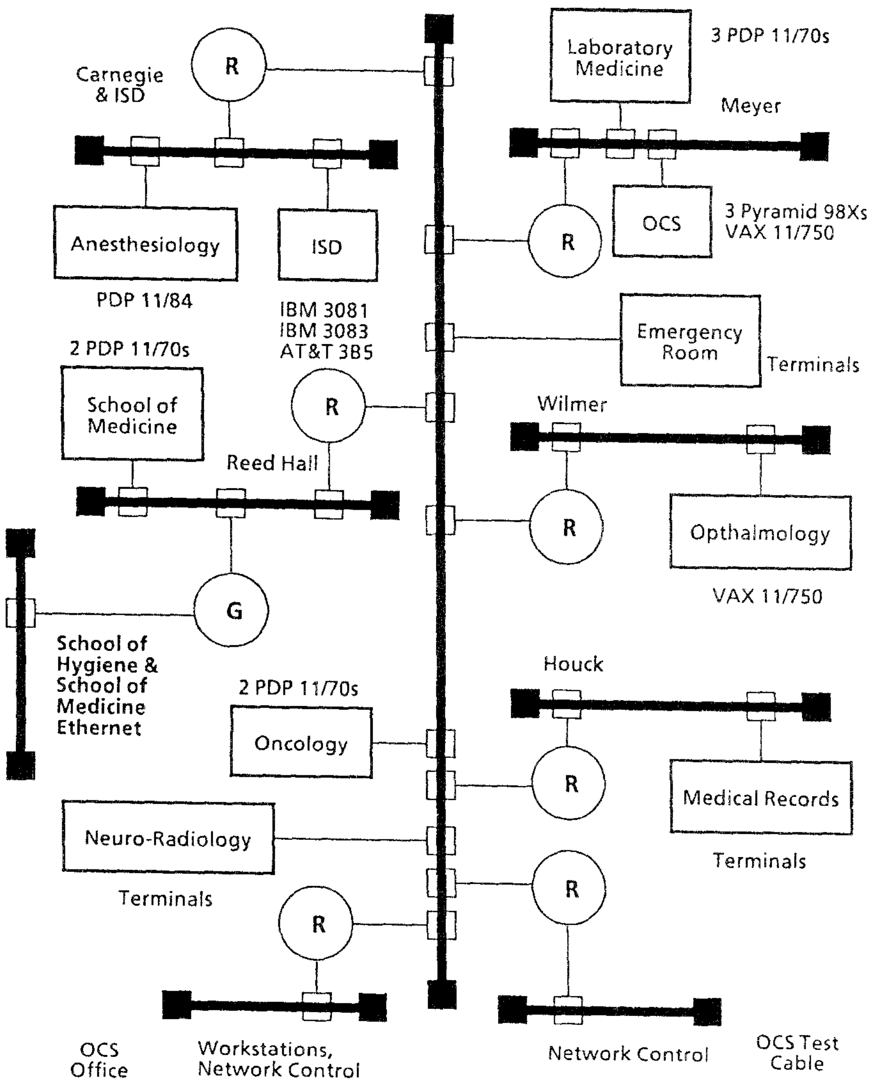

Figure 1. Johns Hopkins ETHERNET network diagram.

depth tree. Earlier work in hospital networking and calculations of expected traffic load have shown that the 10 Mbps band width is sufficiently greater than what is needed to leave adequate room for future demands. Furthermore, band width can be increased in units of 10 Mbps by use of gateway servers.

Multiple vendor equipment can be connected on the same Ethernet cable to form one or more logically distinct networks. These may run the same or different ''high-level'' protocols. Several physically distinct Ethernets can be connected to provide one logical network with an additive total capacity by using off-the-shelf gateway servers or ''local bridges.'' The Ethernet topological restrictions do not prevent its effective deployment in a physical facility as large as, or larger than, the JHH. For example, devices such as host computer centers or terminal clusters can be located as far as 3,000 meters from any segment and connected by means of a fiber-optic transceiver extender. Also, remote segments can be attached to the main segment by fiber-optic repeaters that use fiber cable up to 1,000 meters long; a 500-meter remote coax segment can be connected to the far end of this cable. Several such remote segments may be used. Ethernets across the world can be logically connected through public carrier data networks (PDNs) using inexpensive, commercially available technology.

Ethernet is relatively easy and inexpensive to install in existing buildings. This is essential since much migration of locations is known to occur in the JHH, and network expansion and reconfiguration must occur without any interruption of service. This is in contrast to other technologies such as broad-band coaxial cable, which is expensive to install, requires *much* more expertise to design and maintain, and has no industry-accepted host-to-NIU interface standard as does Ethernet. This means that access to the broad-band medium must be via a particular vendor's ''network interface unit.'' To do this at high speeds requires very complex driver and protocol software development in the host computer's operating system and special (e.g., parallel interface) hardware that may not be always available. To do this at low (RS-232, 9,600 baud) speeds is easy, but it is contrary to the goal of high-speed computer–computer communications.

Reliability is a key concern in hospital operations. All broad-band systems have a ''head-end'' device at the root of the cable plant; this is an active device, and should it fail, network operations will cease. While this device can be duplexed, it is still a single point of failure. Furthermore, broad-band systems require the use of in-line amplifiers, which are active devices and can fail, thus disrupting service to a variable extent, depending on their location. In general, however, broad-band components are very reliable since they are used in the CATV industry. On the other hand, Ethernet does not have a single point of failure. Devices are attached by means of transceivers. If these fail (assuming they are of the vampire vs. in-line variety), only the local node is likely to be affected. Also, these transceivers are extremely reliable. Repeaters are used to connect Ethernet segments if necessary. These are active devices. Their failure will isolate the attached segment from the main cable segment, but both the main segment and its other attached segments along with the detached segment will continue operating as subnetworks.

Expandability is also a requirement. Up to 1,024 nodal addresses (taps) may exist on a given Ethernet and up to 100 taps (including repeaters to attached segments) may be on any given segment. Thus, even without gateways, a very large collection of computers and terminal/print servers may be connected. For example, a given communications server consumes one tap and may support 32 terminals with the equipment JHH is currently using. A host computer also consumes 1 tap, as does a repeater.

There are thousands of Ethernets installed in the United States, so adequate experience exists to assert that the technology is mature and well-understood. For all these reasons, Ethernet has been selected as the basis of the JHH communications network and the Johns Hopkins Medical Institutions Internet.

ETHERNET HARDWARE

The basic transmission mediums are the Ethernet coaxial cable and the transceiver drop cable. Transceivers connect the transceiver cable to the main Ethernet cable. Ethernet controllers permit high-speed data transmissions from host computer to host com-

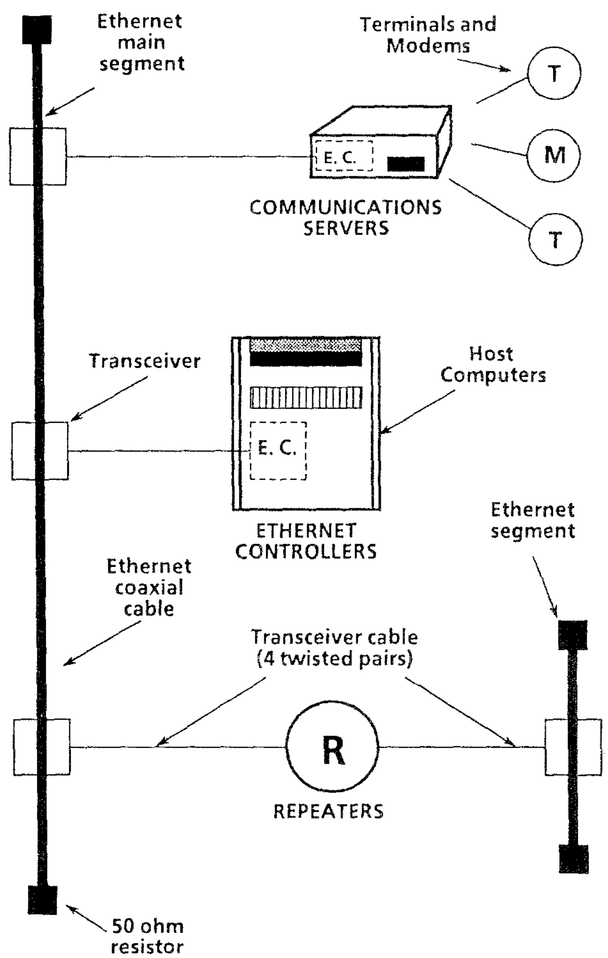

Figure 2. ETHERNET hook-up diagram.

puter. Communication servers are multiport devices that allow network terminals, printers, and other such devices to connect to any host available on the network. Communication servers provide the following additional functions: port contention, clearinghouse and logical name service, port rotaries, call queueing, and network administration.

Cable

Ethernet installations require two types of cable. One is a special coaxial cable (Ethernet cable) and the other is four twisted-wire pairs (transceiver cable). Both cables are heavily shielded. The maximum coaxial cable length for any given segment is 500 meters (1,640 feet). The maximum transceiver cable length is 50 meters (164 feet). The transceiver cable connects the transceiver with the node to be networked. The node might be a communications server (with an integral Ethernet controller) or an Ethernet controller (e.g., in a host computer or interface device).

Transceivers

The transceiver is attached to the Ethernet coaxial cable and encodes and decodes the signals on the channel. Certain signals such as the carrier sense indicate that a transmission is on the channel, and collision detect indicates that simultaneous transmissions have occurred. Transmissions that come in over the Ethernet cable are decoded and sent out on a transceiver cable to the network device.

Communications Servers

Communications servers are used for connecting terminals, remote printers, and modems to a host computer via the network. Typically, communications servers support asynchronous communications interfaces consisting of programmable RS-232 ports. Some vendors support synchronous, IEEE 488, parallel, and other types of communications interfaces. Operationally, the server establishes a virtual circuit between any two ports on the network, thus creating what appears to be a point-to-point connection between them. This virtual circuit can be connected or disconnected under user control.

In addition, a communications server can act like a port contention device. Since the virtual circuit can be connected and disconnected as required, fewer host ports are required to support low-duty-cycle terminals. When a terminal is not connected via a virtual circuit, that host port is available for use by another network terminal. To further the efficient use of host ports, rotary groups can be established. A rotary is a pool of host ports that are automatically cycled through when a network connection is requested by a terminal. An available port in the rotary is then assigned to the terminal. Call queueing is a programmable function that creates a virtual circuit as soon as a port becomes available. The clearinghouse function acts like a logical name server and allows mnemonics to be used instead of Ethernet addresses when making connections. Communications servers also provide some administrative functions for network monitoring and reporting.

The communications server selected for Johns Hopkins Hospital is made by Bridge Communications. It was selected for the flexible programming features, high performance, internetworking architecture support, extensive set of commands, features and utilities, macro support, and quality of design. The server is of sturdy construction, reliable, and easy to maintain. Since the hospital is maintaining the servers internally, ease of maintenance was an especially important feature. The Bridge server also provides network security by being able to request passwords at selectable network access points.

A few special models of Bridge communications servers are used. The Bridge HSM connects to a VAX Unibus under VMS or UNIX. It looks like up to 8 DMF-32s (i.e., 64 ports) to the VAX. The Bridge SNA connects to an IBM host 3705 or 3725 over SNA connections and provides ASCII asynchronous to 3278 synchronous protocol conversion and terminal emulation. Some of the Bridge communications servers also support X.25 interfaces. Gateway devices into X.25 Packet Data networks and between Ethernets also exist. The communications servers use either the Xerox XNS or DoD TCP/IP protocols for communications among themselves over the Ethernet. Use of such industry-standard protocols and the ability to access them (e.g., TCP/IP's Telnet) can provide very flexible communications options.

Repeaters

Repeaters are used to extend the channel length and topology beyond the single main segment. Signals are amplified while passing through the repeater. An on/off switch is available to physically disconnect one segment from another while maintaining the two separate segments. In the event of a repeater failure, the two segments would still work but could not pass information to each other. The repeater selected for Johns Hopkins is made by the Xerox Corporation.

Ethernet Controllers

Ethernet controllers are used for host-to-host computer communications. Controllers must be specifically matched for the type of computer they are installed in and its software. Since the controller board is mounted internally, it must be designed specifically for that computer's bus structure. A device driver must be installed that is compatible with the computer's operating system. The functions the controller supports are the media access protocol and the client interface. Variations in data throughput can be achieved by using either intelligent or nonintelligent versions of controllers. Intelligent controllers permit downloading of the transport control protocols into a board-resident microprocessor, thus reducing host computer protocol processing overhead. This may not always result in greater throughput. The Ethernet controller must match the model of transceiver used. For example, one version controller cannot be assumed to talk to another version transceiver because the signals generated by both may be interpreted differently. However, some controllers are switchable and can handle different transceiver models without any problems.

TRANSPORT PROTOCOLS

A crucial aspect of networking involves the selection and implementation of communications transport protocols, or rules, for machine-to-machine data transfers. The Ethernet does not address this problem. The Ethernet's strength, in fact, comes from the fact that it is dumb and is isolated from higher-level ''client'' protocols. However, reliability is not 100% assured by Ethernet data transfers. The Ethernet implements the physical and link levels of the ISO Reference Model; however, it does not provide a reliable end-to-end service. While collisions are handled and errors are detected, there are resends only on collisions, not on bad packets. The end-to-end integrity is the job of the transport layer protocols.

It is important that a generally standard be used that can be implemented on a wide range of systems from PCs to supermicros to minis to mainframes under a wide variety of operating systems. However, the design must permit computers to access off-loaded protocols if they cannot be implemented as machine resident.

Only two (implemented) protocol suites are nonproprietary accepted standards. These are the DoD military defense network standards (IP/TCP) and the public-domain Xerox Network Standards (XNS). The selection of a protocol suite is a complex matter. It should also be noted that the International Standard Organization has proposed a higher-level protocol suite that, while not currently generally available, may be important in 2 years or so. Thus, applications isolation from the details of the transport mechanism is very desirable. Typically, an application will interact with a protocol that is a user of the transport mechanism. Examples are a remote procedure call protocol, a file transfer protocol, a document representation protocol, or a standardized electronic mail format protocol.

The JHH network team has implemented both the IP/TCP protocols, which are widely supported, and the XNS protocols through the sequenced packet protocol layer (SPP). We plan also to implement the XNS Courier remote procedure call protocol in our UNIX computers, primarily to communicate with Xerox 6085 workstations.

It is possible for all current JHH computers to be networked using industry standard transport protocols, and it is likely that any new computers to be acquired will also be networkable using Ethernet and such protocols. Various personal computers and workstations can also be supported, although our approach is to organize these into separate (Thin-net) Ethernets and bridge or gateway these into the main network as appropriate. Some computers, such as PDP 11's running MUMPS in native mode, present special problems.

BACK-END VERSUS FRONT-END NETWORKING

The JHH Ethernet is being used for two distinct networking purposes. One of these is computer-to-computer communications. This is known as a ''back-end network'' and typically requires a high-speed system. The back-end network at JHH will be used to support the rapid exchange of messages (transactions) between different applications cooperating to provide clinical systems support.

The second network purpose is to provide access to multiple computers from terminals, so that only one terminal, rather than many, will be required when access to several different applications systems is necessary. This is known as a ''front-end'' network. Also of interest is a mechanism to provide sharing of devices such as ''network printers'' among a collection of computers. This is being done to distribute printing functions out of the DP centers and into user areas and to share expensive devices such as laser printers. This may be accomplished as a purely "front-end" function by implementing a "virtual circuit handler" to negotiate front-end virtual circuits using communications servers between host ports and the printer; this software must be integrated with the host print spooler and must permit "fair" access to the shared resources.

The implementation of the back-end network requires Ethernet controller boards for the computers as well as transport software for each system with drivers for the selected boards. While boards exist for all hardware systems in use at JHH, or suitable device attachments exist, software is still a problem for computers that run native MUMPS (i.e., PDP 11s). Although Ethernet Unibus controllers exist, neither MUMPS driver software nor transport protocol software to support Open Systems Interconnection exists.

Since these systems can only communicate at 9,600 baud by passing ASCII character streams through asynchronous terminal I/O controllers (RS-232 ports), and since the network team had experience with character loss when this data rate was sustained (e.g., on DZ controllers), we developed a special RS-232 interface protocol. The MUMPS (and any other such) systems must communicate to a device that off-loads the transport protocols and high-speed Ethernet interface. This communications is subject to error checking link protocol on the RS-232 lines. Further, a command interface is defined for opening and closing network connections to logically identified destinations (servers).

We developed two approaches based on different protocol surrogate devices and network architectures to implement reliable 9,600-baud communications. Approach one involved programming the Bridge CS-100 server and connecting MUMPS ports as user and server ports to the "Bridge box." This is a fairly inexpensive approach and fully distributes the MUMPS support, so it is reliable. However, programming the box proved to be difficult, and we felt we might be sensitive to future releases of the Bridge software and thus have a maintenance problem. Therefore, we developed an RS-232 gateway, called the cluster machine (CM). One or more UNIX machines are used as a central node. All MUMPS systems are connected over permanent network (Bridge front-end) virtual circuits to the CM's RS-232 ports using a reliable end-to-end protocol.

The IBM mainframes present an interesting back-end connection problem. Our solution is described below.

TRANSACTION PROCESSING

It is insufficient to provide a communications medium and set of compatible high-level protocols (e.g., transport protocols) without understanding how distributed applications may be built. There are several *models* of distributed data processing that exist. An ideal model is to have complete transparency to the applications and user of where data, programs, and other resources are located and how they are remotely accessed. In this scheme, a distributed operating system exists and mediates all requests for services, perhaps at a slight cost in performance for remote task execution.

In the JHH environment, this scheme is not possible due to the heterogeneity and incompatibility of the diverse system software (operating systems). Rather than have operating systems directly communicate with each other, distributed functionality can be built by having applications exchange well-defined messages among themselves across computer boundaries at high speeds. This is called a *transaction processing model* and the messages among the applications are called transactions. (*Transaction* is an overly used word in computer science; it has one meaning in the world of IBM software, another meaning in data base programming theory, and yet another in distributed systems theory.) Transactions may be prescheduled, event-driven (e.g., an admission causes a message to be sent to a computer that handles inpatient management providing relevant admit details, location, etc.), or demand-initiated (e.g., a clinical unit such as ER sends laboratory test order transactions to the Clinical Laboratory system). Also, transactions may be short messages or lengthy file transfers (e.g., the transfer of surgical pathology reports to the automated clinical resume discharge system). The semantics of a transaction may be a one-way message or a two-way request/response session (e.g., patient identification request from a clinics system to a PID system and response set of appropriate information on matching patients).

Two general kinds of services need to be implemented on top of the transport protocols for applications support. First, there must be a way for a program to access remote resources or for a service to be accessed by a remote user. Second, there must be a bulk file transfer capability that can resolve record and file structure and description incompatibilities among different operating systems. The IP/TCP related protocol, FTP, can be used for this function, or as a model. A simpler model is the BSD UNIX networking utility rcp. For the first problem, a "remote procedure call protocol" is needed.

Approach

To support transaction processing, the *remote procedure call* (RPC) model was adopted (see Figure 3). A RPC protocol provides access to services on remote devices transparently to local applications by making the access look like a local subroutine call. In reality, the subroutine is distributed with a piece on each computer. The application is

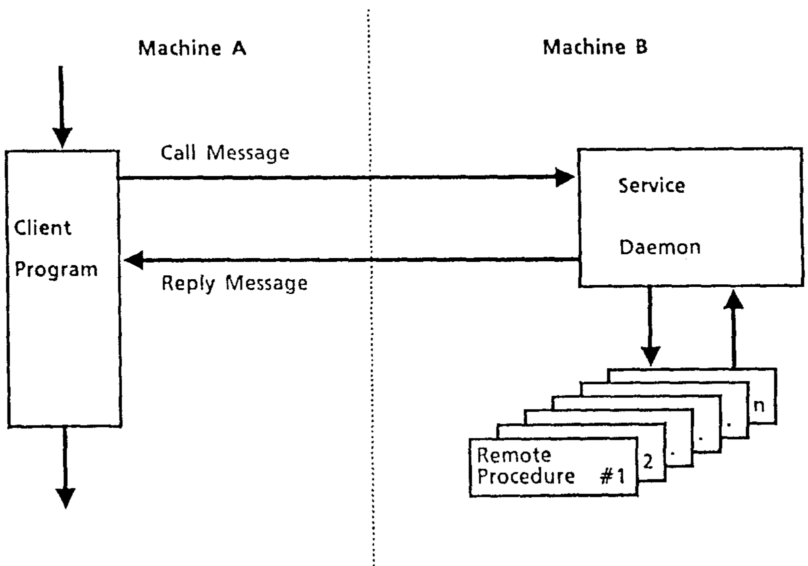

Figure 3. Interprocess communications with remote procedure calls.

shielded from complex network and translation details, as well as the details of server structure or logic. While quite useful, this model handles only a subset of distributed processing problems. To implement RPC, data representation differences need to be resolved across different machine architectures and the syntax and semantics of the rpc need to be defined. The Sun RPC/XDR protocol is designed to handle these problems. There are two standards for RPC, the Sun Microsystems RPC/XDR (external data representation) protocol developed to support the Sun-distributed Network File System, and the Xerox Courier protocol. To date, we have been using the Sun RPC in the UNIX environments and have implemented a subset of this under MVS/CICS for transaction processing with the IBM mainframes. We have also implemented a limited subset of RPC for MUMPS systems. We intend also to employ the Courier protocol in the future, depending on our workstation decisions.

Two interesting problems required resolution. The IBM mainframes needed to be connected to the Ethernet at high speeds and enabled as reasonably full network citizens. Also, the MUMPS systems needed to be brought into the network scheme. Both of these tasks have been accomplished. In addition, access to the mainframe from standard ASCII asynchronous terminals via 3278 emulation is supported from any networked vt100-like terminal.

Certain architectural decisions were made that parallel the "client" and "server"

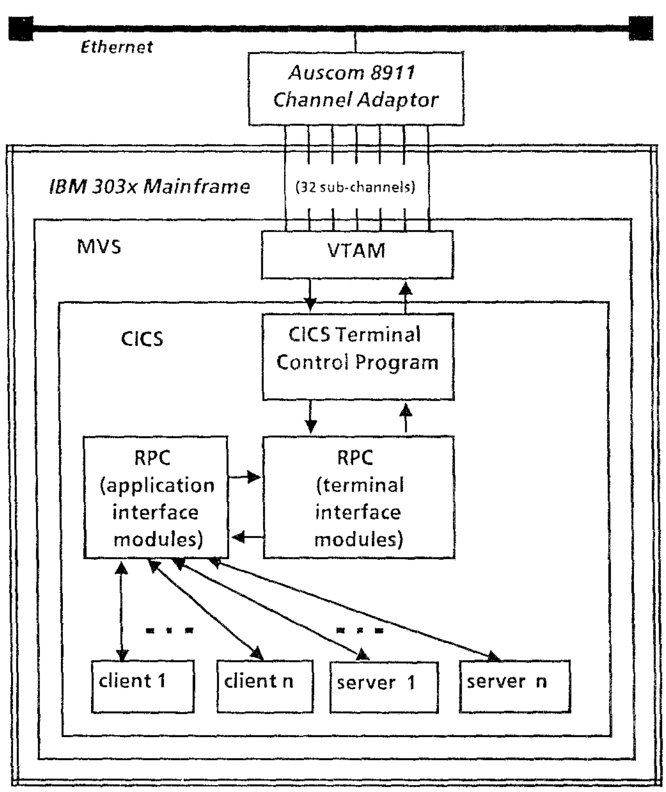

Figure 4. ETHERNET-to-IBM/CICS architecture for remote procedure call interface.

concepts of the RPC model. We regard the distributed system to consist of client (or user) components and server components. Thus, (for example) a large central data base is a server logically (and many times physically) separated from the client applications systems or end-users. The clients may be different computer systems supporting end-users or workstation devices. The clients know no details of the server design (and v.v.); they interface via a specification that is an RPC.

Solutions

Integration of the PDP MUMPS systems required a more difficult and less satisfactory solution than the IBM mainframes. While the PDPs can support Ethernet controllers, there is no existing operating system software to permit use of the hardware in an Open Systems Interconnect architecture. Since the MUMPS product is proprietary, we were not able to develop system software. Until this situation changes, our approach involves use of an "RS-232 ASCII" gateway. Basically, we developed software in the UNIX superminis that acts as an intelligent gateway connecting multiple MUMPS systems to all the other Ethernet-attached systems as well as to each other. A reliable link control protocol was developed and implemented between host ports on the PDPs and the UNIX Pyramid systems to assure that characters would not be lost and that noise errors would be detected. Since the MUMPS PDP systems are physically remote from each other and from other systems, the Ethernet was used for connectivity by means of permanent virtual circuits between ports of the Bridge communications servers. The UNIX gateway software performs additional RPC/XDR translations (known as serialization and deserialization) from the MUMPS ASCII data records into the network standard data type representations. This approach is quite general and could be used for any systems supporting RS-232 ASCII communications.

For the IBM mainframe, two problems existed. First, it was desirable to establish high-speed interfacing between the Ethernet and the computer. Second, it was necessary to develop RPC/XDR software and applications interface methods. The solution of the first problem introduced some additional problems, however. We employed an Auscom 8911 channel adapter which attaches to a mainframe channel (almost) emulating a standard 3274-1D controller on one side and which has an Interlan Ethernet controller on the other side. The 8911 is an LSI-11 Q-bus machine running a proprietary operating system.

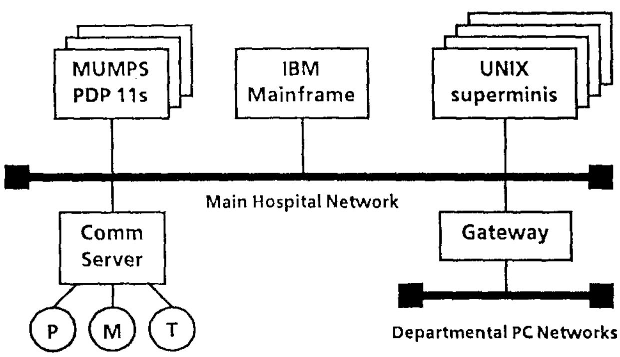

Figure 5. Simplified schematic of JHH ETHERNET network.

Auscom supplied an XNS/SPP package for the device and, at the request of JHH, made several improvements to the product. The device appears to CICS like a collection of 3278 terminals that are accessed via VTAM. However, since host client and server functions needed to be implemented, interaction with the 8911 was needed to "open the virtual terminal" for client use. A significant problem was that we were using IP/TCP protocols among UNIX systems but needed XNS to communicate with the IBM through the Auscom. XNS implementations tend to be less mutually compatible than TCP at this time. Pyramid Technology supplied us with an XNS/SPP package in their UNIX kernel which we were able to make work with the Auscom implementation. We then implemented the Sun RPC/XDR protocol on top of XNS as well as TCP in the Pyramids. Also, we implemented the XDR string and integer data type subset of RPC/XDR under CICS/MVS and defined the application interface. This enables high-speed functional integration of the mainframes with the rest of the computing environment, so the mainframe applications could access remote data bases and could provide services to the other systems. This capability is in production use for remote data base access and systems integration.

SUMMARY

The Johns Hopkins Hospital has developed distributed processing solutions to many common problems arising from the independent development of heterogeneous computers employed to solve departmental problems. A highly distributed, but functionally integrated set of clinical information systems is resulting from a strategic architectural plan. Considerable progress has been made to provide the necessary technical support for functional systems integration.