NASA CR- 177853

(NASA-CR-177853) STATE OF THE ART SURVEY OF NETWORK OPERATING SYSTEMS DEVELOPMENT (Computer Technology Associates, Inc.) 224 p HC A10/MF A01 CSCL 20F

N86-21352

Unclas G3/74 15545

COMPUTER TECHNOLOGY ASSOCIATES, INC.

Denver • Washington, D.C. • Colorado Springs • Albuquerque • San Jose

STATE OF THE ART SURVEY

OF

NETWORK OPERATING SYSTEMS

DEVELOPMENT

Performed for

NASA Goddard Space Flight Center
Code 700
Greenbelt, MD

CONTRACT NO.: NAS5-28583

By

Computer Technology Associates
10227 Wincopin Circle, Suite 208
Columbia, Maryland 21044

June 7, 1985

TABLE OF CONTENTS

1.0 INTRODUCTION

In this report Computer Technology Associates (CTA) presents the results of its State-of-the-Art Survey of Network Operating Systems performed for Goddard Space Flight Center. In this section of the report CTA states the objective of the study, presents the major results of the study, and describes the organization of the report. The objective of this report is to determine the current state-of-the-art in Network Operating System (NOS) development. CTA has focused this study on Local Area Networks (LAN's) in order to relate the NOS developments to LAN's aboard the U.S. Space Station . An on-board data system will coordinate the activities of several space station subsystems and many users while providing access to commonly used data. It is envisioned that the NOS will be key to integrating these activities as well as handling requests for ancillary data for experiments and subsystem monitoring.

CTA has drawn the following major conclusions as a result of this study:

- o Developers are primarily working in either UNIX or the IBM PC DOS environments. Few vendors are providing interoperability between the UNIX and PC DOS environments. Recent announcements by commercial vendors indicate that products with UNIX-PC DOS file service will soon be available.
- o Services offered most commonly by vendors are file service and print services; also many vendors provide messaging (electronic mail) and remote procedure call services.
- o Major deficiencies are the lack of data representation service and network control service for handling failure conditions.
- o Interprocess communications services, in many cases, provide inadequate performance and are not provided for PC based systems. A number of research activities have

demonstrated efficient interprocess communications.

- o Portability of the NOS is a major problem because of
  - no standard local OS
  - integration of NOS and Local OS
- o UNIX is a potential solution to the portability issue, but it has not been standardized and has serious deficiencies in interprocess communication and real time I/O.
- o Open Systems Interconnection (OSI) reference model and protocol standards are not being implemented because vendors have focused on maximizing performance using simpler protocols with fewer layers in LAN enviroment.
- o As standards evolve and there is a greater requirement for internetworking the use of these standards will become necessary and more common.

To address the state-of-the-art of NOS development, CTA has partitioned this report into five sections and three appendices. In Section 2, CTA presents its methodology for acquiring, organizing, evaluating, and documenting information on candidate commercial and research NOS packages. CTA has identified literature sources utilized for the review and evaluation process, the organization of all material by means of computerized data bases, and the use of a template to provide concise, individual descriptions of commercial and research NOS's.

In Section 3, CTA presents an overview of Network Operating Systems to establish a common terminology and technology areas to be addressed in the survey. The contents of this section include:

- o description of a network operating system,
- o relationship to a distributed operating system,
- o relationship to the ISO Open Systems Interconnection Reference Model,

- o assumptions stating specific types of products or systems, though related to network operating systems, do not qualify as network operating systems and are not addressed in detail in the survey.

Work relating to network operating systems is being performed by two research communities: computer networks and distributed processing (and database). The former group deals primarily with loosely coupled systems and builds the network operating system upon the local host operating system. This group has developed and sponsored the OSI technology. The latter group deals with more tightly coupled systems and the networking features are closely integrated with the local operating system.

Although the work of the computer network research community is closer to the proposed thrust of this study, CTA has attempted to cover both areas. This is necessary because the separation of the two groups is ambiguous. CTA believes that ultimately the work of the two groups will converge such that the communications will simply appear as another peripheral to the local operating system.

The results of this report will be input to a subsequent effort to develop a prototype Network Operating System. Thus, to expedite this development, CTA has focused both on development of commercial products and research and development efforts. In particular, the prototype NOS could be more cost effectively developed building upon an existing product rather than developed completely anew.

In Section 4 a set of NOS functional characteristics is presented in terms of user communication, data migration, job migration, network control, and common functional categories.

The common category includes those functions which are common to any two or more of the other categories.

In Section 5, CTA presents the results of its NOS survey, which considered products (current or future) as well as research and prototyping efforts. First the key trends and observations are presented. Then a brief narrative is provided highlighting the important characteristics of Network Operating Systems. Narratives are provided only for selected NOS based on the following criteria:

- o relative importance in field based on number of installations,
- o introduction of new or advanced technology,
- o completeness of functional capability.

In the last section (Section 6), CTA presents an evaluation of the NOS products which are relevant in the space station environment and its activities. Features provided as well as deficiencies are described and presented. Then the required features are prioritized to facilitate development of an implementation plan.

CTA has also included three major appendices to this report. They are:

- o a set of product templates,
- o a bibliography of relevant literature,
- o a bibliography of relevant standards, and
- o a bibliography of research and development efforts.

2.0 METHODOLOGY

In this section CTA presents its methodology for acquiring, organizing, evaluating, and documenting information for candidate Network Operating System. In the survey, commercial and research products currently available or under development were included. Tabulated results differentiate between those products which are currently or soon-to-be commercially available versus research projects in the academic or government environments.

CTA's methodology to survey the available NOS products and projects involved the:

- identification and acquisition of relevant information sources,
- review and assimilation of relevant articles, product announcements, etc., from these sources,
- create computerized data bases with product templates to concisely describe it,
- generation of computerized data bases to organize the information gathered during the survey, and
- review and analysis of vendor or developer provided information
- comparison of the features provided versus Space Station requirements

The initial step in the survey was to define a Network Operating System and establish to relevant products and systems to be surveyed. In particular there are many products, and systems, though related to the NOS technology, do not qualify as a NOS. These were identified, but not studied in detail.

In defining a NOS, CTA distinguished between network and distributed operating systems. These concepts are very closely related with certain systems and products categorized in a gray area between a network and distributed operating system. Thus both network and distributed operating systems were considered.

The major sources of information for identifying candidate network operating systems were:

- o professional journals published by the IEEE and ACM,
- o Trade journals produced for the engineering, computer, and telecommunication industry (e.g., "Computer Design," "Systems and Software," "Mini-Micro Systems," and "Telecommunications Products Plus Technology"),
- o PC-oriented monthly magazines (e.g., "PC-World", "PC"),
- o Relevant product news releases,
- o NASA specific information; in particular work done by Dr. Edwin Foudriat at NASA Langley.

In the first area the recent papers by J. A. Stankovic (and his colleaques in the latter paper)

> "A Perspective on Distributed Computer Systems", IEEE Transactions on Computers, December 1984

> "A Review of Current Research and Critical Issues in Distributed System Software", IEEE Distributed Processing Technical Committee Newsletter, March 1985

was especially helpful in identifying candidate systems, especially in the research environment, and assessing the state-of-the-art for key issues.

Also the January 1985 Workshop on Operating Systems in Computer Systems in Computer Networks hold in Ruschlikon, Switzerland provided a very recent summary of current work in this area. This workshop was organized by Liba Svobodova and session summaries are presented in the ACM publication:

> Operating Systems Review, April 1985

In addition, the very recent conference proceedings; 5th International Conference on Distributed Computing Systems was briefly reviewed.

The trade publications were useful for assessing the capabilities of current and future vendor products. The NASA specific information was useful for relating the NOS technology to space applications.

The assimilation and organization of the NOS information was facilitated by the generation of a NOS template consisting of the key characteristics of a NOS. This format of this template is illustrated in Figure 2.1. CTA filled in the relevant attributes in the template for each system or product considered, although not all information was available. The resulting set of templates were physically created and stored on an IBM Personal Computer using Lotus 123™ (TM of Lotus Development Corporation). For research projects, a less detailed template was generated. The information included in this template consists of project name, organization, contact, and key words.

CTA then synthesized the functional characteristics for a Network Operating System based on its assessment of:

- o features implemented in vendor products or proposed for future vendor products,
- o features implemented in university, government, or industrial research projects,
- o features defined in current or proposed standards.

This set of functional characteristics is presented in Section 4.

The analysis and review activities performed by CTA involved identification of technology trends, and comparison of the state-of-the-art technology versus space station requirements. The capabilities of the candidate systems have been reviewed relative to:

- o completeness of functionality,
- o adherence to OSI protocols,
- o portability and performance.

The space station requirements were derived from the requirements analysis performed by CTA for its Fiber Optic Demonstration System High Level protocol study.

Having codified requirements and assessed the state-of-the-art in NOS development, CTA then prioritized NOS features for implementation. This prioritization was based on relative importance of the feature, implementation complexity, and technical maturity.

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	
2	Developer/Vendor:	
3	Address:	
4	Telephone No(s))/Contact:	
5	Documentation Availability/Source:	
6	NOS Hardware Implementation: - Network: - Host Computer(s)	
7	Host Operating System Name(s):	
8	Implementation Language:	
9	Maintenance Support:	
10	ISO Model Implementation:	
11	List of Services for Each ISO Layer:	
12	Major O/S Functions Implemented:	
13	O/S Modification(s):	
14	Price:	
15	User Added Applications S/W:	

Comments:

FIGURE 2.1
Product Template Format

3.0 OVERVIEW

3.1 Description

The initial effort was to define a Network Operating System and to establish its relationship with other concepts in the distributed computing environment such as distributed operating systems and Open Systems Interconnection. A NOS can be viewed as a collection of Application Servers and communication software needed to provide User-to-Server, User-to-user, Server-to-Server communication. A server is a computing process which performs a specific function at the request of a client, usually a user-process. Servers may rely on specific hardware peripheral such as disks, printers, and modems or a server may be strictly a computational server or process, e.g., a user may send a command to another node for remote execution of a program.

The resulting definition of a NOS is:

"A Network Operating System provides the capabilities of:

- Reading/Writing information on remote devices,
- Sending/Receiving information to/from remote processes or users, and
- Executing/Controlling jobs on remote devices

as if they were local."

The information exchanged may be textual or non-textual. Two characteristics which are important in defining a network operating system and distinguishing it from distributed operating systems are the:

- o interface between the network operating system and the local operating system
- o level of transparency provided to users of the network operating systems.

These characteristics are described below:

Network Operating Systems employ the services of the local host operating system to perform these functions, as illustrated in Figure 3.1. The NOS environment is homogeneous when the local operating system is the same at all sites. This facilitates portability of the NOS. If the NOS and the local operating system are integrated, then the resulting operating system is referred to as a distributed operating system. The degree of integration is subjective, and both network and distributed operating systems are addressed in this survey.

In an ideal situation the NOS would reside between the local operating system and the applications. The NOS would appear as an application to the local operating system and as an operating system to the application. With this implementation the NOS could be easily ported to other hosts employing the same operating system. The NOS would handle system calls by the applications and determine if the system call must be satisfied by local or remote operating systems. If the system is handled locally, then it is passed to the local operating system. If the system call is handled remotely, then it is passed through the NOS to the Network I/O driver for delivery through the network to the remote system. Note that the NOS may not handle all calls, e.g., clock and memory management services may have only local significance.

The local operating system may also be distributed itself because the local host may be a multiprocessor device where the processors are distributed over a device backplane bus. Such distribution of a tightly coupled system is not considered in this study.

The Network Operating System employs the services of the local operating system for a wide number of services such as interprocess communication (among processes in the local host), memory management, clock, and file service. Although this implementation facilitates portability, it introduces significant performance problems. To achieve high performance, developers integrate the NOS and local operating system reducing the portability. The associated tradeoffs are discussed in Section 6. One of the major goals of a Network Operating System is to provide transparent operation to the user, which may be either a person or process. Network transparency can be achieved at four levels which introduce corresponding level of system complexity:

- o name transparency in which each object and resource has a name independent of the site from which it was invoked as part of a system command,
- o location transparency in which the location of an object or resource is not encoded in its name,
- o semantic transparency in which the commands employed to invoke NOS services are independent of the location at which they are invoked,
- o performance transparency in which the system response time to a service request is independent of site at which the service request is executed.

Name transparency is a necessity for both network and distributed operating systems because without it there would be enormous complexities in moving or distributing software. For example as applications software is ported to new hosts any object or resource names employed in the software would take on new meaning when the software is executed on the new host.

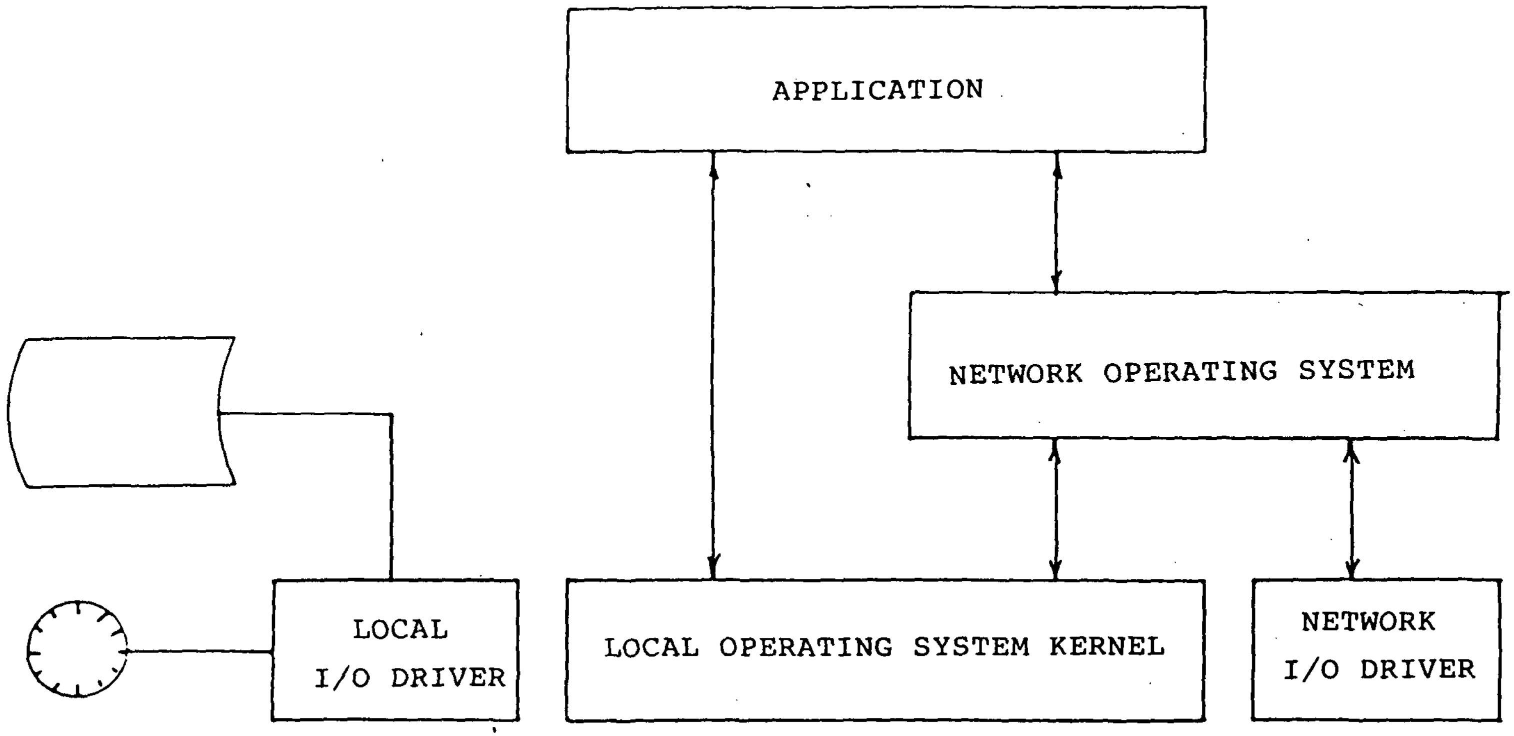


FIGURE 3.1: System Architecture

Complementary to naming transparency, location transparency is a desirable feature, but probably not necessary for a NOS. For example, it is not unreasonable to define a pathname for an object such as a file in which the location of the object is prepended. Resources could be similarly handled with pathnames. However since resources are not frequently created or change status, they could be given names without encoded locations. Location transparency for a distributed operating system is a much more important objective because local and remote resources are much more integrated. For example in some systems, the user is not aware of the locations of a resource or object when a service request is made. The system is responsible for determining where the resource or object request is located. If there are multiple resources or objects capable of satisfying the request, then the system must select (via some optimization criteria) which one should provide the service.

In summary, for a network operating system, objects such as files will have the location name prepended to the file name. For resources this may also be acceptable, but it is not especially complex to use names without the site name encoded.

Semantic transparency is a necessity for both a NOS and DOS. Without such transparency, porting and maintaining software would be extremely difficult. Furthermore, a common set of commands will facilitate use of the system by users who utilize more than one host.

Performance transparency is a desirable objective for both a DOS and NOS, but clearly the performance of the computer network linking the hosts will establish a bound on how well this objective can be achieved. With its tightly integrated architecture, performance transparency is more critical for a DOS. However with current technology advances providing increases in processing speed and communications transmission speed, this objective can be more nearly achieved with a NOS.

3.2 Relationship with the Open System Interconnection Reference Model

The major standard relating to a Network Operating System is the ISO Open System Interconnection (OSI) Reference Model, which is an architectural model for communications between two open systems. The OSI Reference Model defines the protocol functions to be performed by seven protocol layers. In addition, ISO is defining (and has defined) specific protocols for each of the seven layers of the Reference Model.

These protocols define:

- o peer to peer communication roles between corresponding layers in communicating open systems,
- o primitive interfaces between adjacent layers in the same open system. For a layer protocol, the interface is between layers N and N+1, as well as layers N and N-1. For N equal 7, the N to N+1 layer interface is the interface to the user.

However, these protocols do not define how the protocol is to be implemented in the local open system. For example, buffer management and interface to the operating system in the local open system are not defined by these protocols.

An implementation of these protocols in actual hardware and software is a network operating system. However, a network operating system does not have to adhere to the Reference Model or ISO OSI protocols.

There are fundamental misconceptions about the OSI model regarding subsequent implementation of OSI protocols. The OSI Reference Model defines an architecture and OSI protocols provide the detailed interface specification between open systems. Nothing is specified regarding the implementation of the protocols in the local open system. For modularity, it would be nice to implement these protocols as separate tasks. However, this introduces substantial overhead with buffering and local interprocess communication. OSI imposes no constraints on how the protocols are implemented, hence the developer of the local system could implement two or more layers in the same task to achieve higher performance at the expense of modularity. As long as the OSI protocol suite appears to the outside world as defined in the specification, no interface problems occur.

3.3 Assumptions

There are a large number of vendors who have or claim to have a network operating system or provide some features. In the survey CTA has selected vendors based on:

- o relative importance in the field based on a number of installations,
- o introduction of new or advanced services,
- o completeness of functional capability.

In particular, vendors providing only a disk server as opposed to a file server were excluded. A Disk Server only provides service pertaining to:

- o one or more disk sectors,
- o entire volume.

However, a File Server allows users to share individual files rather than just share disk volumes. This distinction has significant implications for a NOS. A typical disk server request will be a "Write Track X, Sector Y on disk volume Z." The client side of the software controls the file structure and directory information. Therefore, the disk server cannot control or resolve conflicts between different users of a disk volume. For example, two different clients may write the same sector and thus leave the file or even the volume directory in an undefined state. In some cases a NOS may provide synchronization tools such as semaphores; however, use of these tools requires development of NOS specific applications.

A typical File Server request is "Open/Close a file," "Read/Write next record," "Read/Write record number n" or "Update record with key X." The file server controls the file structures and the volume directory. Therefore, it knows what files are open rather than which disk blocks are in use on a particular volume. As such it can allow multiple clients to share a file and synchronize their requests.

Appendix A, which lists the NOS product templates, identifies products that provide a Disk Server, but these systems have not been analyzed in detail.

4.0 NOS Functional Characteristics

In this section CTA synthesizes the functional characteristics for a Network Operating System based on its assessment of:

- o features implemented in vendor products or proposed for future vendor products,
- o features implemented in university, government, or industrial research projects,
- o features defined in current or proposed standards.

The purpose of this section is to define the functional requirements for an ideal NOS. It is not expected that:

- o any one vendor would provide all of these features,
- o all of these features are required in the NASA environment.

The features provided by individual vendors are discussed in Section 5, and the features required in the NASA environment are discussed in Section 6.

4.1 User Communication

For user communication service, the Network Operating System may support interuser communication and user-to-system communication. Users may be able to send one another electronic mail, communicate back and forth in a dialogue, and arrange interactive conferences among members in a larger group.

4.1.1 Message Handling Service

The Message Handling Service provides for the store and forward delivery of electronic mail with content of:

- o text,
- o facsimile,

- graphics,
- voice,
- arbitrary binary data,
- any combination of the above.

It includes the capabilities for: entry of mail by the user, transfer of mail between electronic mail centers, delivery of mail to the user, scan in and out boxes, message retrieval and probing the status of messages.

The services provided include a variety of services as illustrated in Figure 4.1 Some services may provide additional features specifying the format of the message, such as the CCITT Document Content Specification to assist in preparation of messages, e.g., establish message format, etc.

Relevant standards are the CCITT X.400 message handling service standard are ISO work in process on Office Document Architecture and Text Preparation and Interchange.

4.1.2 Print Service

The Print Service provides the capability of outputting a hardcopy of electronically stored information at a remote location. The key elements of the print service are:

- spooling of print requests such that a request is queued if the printer is busy,
- real-time service, scheduled or batch service,
- forms control such that specific paper and format is employed,
- fault recovery.

Recovery consists of being able to complete the printing task in progess and then continue with remaining print tasks in

TABLE I

MESSAGE TRANSFER SERVICES

Service Elements

Submission and Delivery
- Message Transfer
- Message Identification
- Submission Time Stamp
- Delivery Time Stamp
- Grade of Delivery Selection
- Deferred Delivery
- Deferred Delivery Concellation
- Non-delivery Notification
- Delivery Notification
- Prevention of Non-delivery Notification
- Multidestination Delivery
- Disclosure of Other Recipients
- Alternate Recipient Allowed Request

Query
- Probe

Status and Information
- Hold for Delivery
- Alternate Recipient Assignment

Conversion
- Content Type Registration
- Original Content Type Indication
- Content Type Conversion Prohibition
- Implicit Content Type Conversion
- Content Type Conversion Indication
- Explicit Content Type Conversion

TABLE II

INTERPERSONAL MANAGING SERVICE ELEMENTS

- Primary and copy recipient indication (i.e.,to:,cc:)
- Blind copy recipient indication (i.e.,bcc:)
- Authorizing users designation (nonlegal signature)
- Subject indication
- User-message identification (personal, private, company confidential)
- Importance indication
- Cross reference indication
- Expiry date indication
- Obsoleting indication (i.e.,supercodes a previous message)
- Reply request indication (i.e., to whom and by when)
- Replying user-message indication (this is in reply to)
- Multipart body
- Body part encryption
- Forwarded user-message body part
- (Non-)receipt notification

Reference: "Message Handling Systems and Protocols", Proceedings of the IEEE, December 1983

TABLE 4.1: X.400 SERVICES

the queue. Completion of the task in progess when a failure occurs (such as printer out of paper, end-of-ribbon) often requires reprinting the last few pages (rather then reprinting the whole file which can be time consuming and wasteful) Unlike a disk failure,sometimes a printer failure cannot be detected in the host computer such as a paper jam or faded ribbon. This implies the need for manual intervention and recovery procedures.

If the device malfunction cannot be repaired, than the NOS may also attempt to route the print request to an alternate device.

4.1.3 **Terminal Handling Service**

The Terminal Handling Service provides the capability of presenting/accepting data to/from the user at his local terminal. This includes the capabilities of:

- o presenting textual (formatted and unformatted) data and nontextual data (facsimile, voice, graphics),
- o displaying the data in window in the user display,
- o communicating terminal to terminal.

The key issues regarding the terminal handling service are:

- o with current technology trends, terminals are really hosts rather than non-intelligent devices, thus they must be able to handle multiple tasks,
- o implementation of a terminal handling service as a virtual terminal or as a parameter driven device; this is critical for environments in which there are many types of terminals,
- o handling non-textual devices as graphics and image applications become more important.

Although there is active standardization work in this area on virtual terminals, it does not address current requirements for exchange of non-textual information.

4.1.4 Conferencing

The conferencing service provides the capability of on-line multiparty user communication. In this service users would enter messages at their workstations and the NOS would display them on the workstations of the other participants.

The key issues in the conferencing service:

- o screen formatting of the user workstation,
- o maintenance of a conference record,
- o templates for specific primitive action items resolutions,
- o protocol for enabling users to enter/leave the conference.

4.2 Job Migration

When a job can be broken into a group of cooperating processes, it is desirable that the Network Operating System be capable of scheduling these processes to run across multiple nodes. Concurrent processing and synchronization are included in this service category.

4.2.1 Remote Job Execution

The Remote Job Execution service can be executed at two levels of sophistication. An elementary capability is to provide a service which permits a user to execute a job on a remote computer simply by invoking a procedure call. A more sophisticated capability is to allow the user to define a set of input sources, processors for execution and output destination. The first alternative is referred to as the remote procedure call service while the latter is referred to as the Job Transfer and Manipulation Service. Both are discussed below.

The Remote Procedure Call Service provides the capability to execute programs on remote computers using a procedure call as if the program were resident on the local computer. This service is sometimes referred to as the "interactive batch" service in that the response time of the service is usually very rapid. This service is analogous to current batch service except results are returned in real time. To submit a request to the Remote Procedure Call, the user invokes a call to a procedure as if it were local and supplies the necessary input parameters and, if necessary, the data structures for return of the results to the originating process. The Network Operating System is responsible for locating the program, establishing the remote connection, implementing the remote procedure, and delivering data to/from it. Depending upon the implementation, the originating process may wait for the results of the procedure call or continue concurrent processing while awaiting the results.

Various other application layer protocols may also employ this service. For example, the Message Handling service proposed by CCITT employs a Remote Procedure Call to distribute electronic messages. The Remote Procedure calls protocol is defined in CCITT standard X.410.

In this terminology, the local process is connected to the remote process via the remote procedure call upon demand during program execution. After execution of the remote procedure and return of results, the local and remote processes have no association. Other types of remote procedure calls bind the local and remote processes during compilation or load time. This connection is maintained throughout the lifetime of the connection. This latter type of remote procedure call is more like the interprocess communication discussed below. In particular, the real-time performance of this protocol will be better because the association is pre-established.

The key issues associated with the RPC are achieving adequate performance with the requisite set-up procedures and handling exception conditions. For example, if the return from the remote procedure does not occur, what action should the application take. For example, the remote procedure may have successfully executed but the response was lost. Simply re-executing the procedure would introduce an error if repeated execution of the remote procedure does not generate the same result. For example, adding $100 to the balance of a bank account does not generate the same result, i.e., the first time through, $100 is added, but if it executed twice, $200 would be added. These problems associated with remote procedure calls have resulted in the categorization of "at least once" or "exactly once" semantics.

The Job Transfer and Manipulation Service, a generalization of remote job entry protocols, provides the capability of the user of specifying how the:

- job should be partitioned into subjobs to be run on different processors,
- results should be processed and controlled by the computers performing the job,
- results should be retrieved and reassembled.

This service provides the distributed computing service in that jobs can be distributed and assigned to the optimal computer for execution. Optimality can be defined in terms of:

- o suitability of a particular computer for performing a job, e.g. scientific computing on a "number crunching" computer,
- o optimizing communications and computer costs,
- o assigning jobs to under-utilized resources.

The relevant standard for Job Transfer and Manipulation is ISO DP 8832.

4.2.2 Interprocess Communication Service

The Interprocess Communication Service provides program to program communicaton between remote process. Its key features are data transfer and process synchronizaton. The interprocess synchronization is a control mechanism which enables remote processes to:

- o effect process precedence control, e.g., determining which process currently has access to a resource,
- o effect shared resource control.

It is analogous to the semaphore in conventional operating systems.

The major issue is providing adequate throughput performance across the network. This requires use of a message oriented communication rather than byte oriented (as in UNIX pipes). Also to enhance performance, it is desirable to notify the receiving process that a message has been received from it rather than requiring the process to keep checking to determine if a message has been received.

The communications primitives provided will have a major impact on performance. A categorization of these primitives based on work by Stankovic [48] is illustrated in Figure 4.2. Using a synchronous send, the user is blocked, i.e., must wait until the message is received by received; thus the source and destination are synchronized. If the no wait send is employed, the user can initiated other tasks, but the NOS must provide buffering to accommodate the message. If a conditional send is employed, then the receiver is blocked waiting for the source message.

When the receiver is blocked awaiting messages from remote locations, either restricted unconditional in which messages from specified processes will be accepted or blind unconditional in which all messages will be accepted. Alternatively in the conditional receive, the communications channels are polled.

In summary, the type of primitive will drive the level of concurrency that can be achieved and limit the performance.

A major application for this service is in distributed control. In this case it may become significantly more complicated when more than two processes are involved. The issues of updating data structures are analogous to updating distributed data bases and are discussed below under Concurrency and Synchronization.

Although no international standards organization is promulgating a standard for interprocess communication, IBM has published its standard, Logical Unit (LU) 6.2 [61]. The key features of the process-to-process communication provided by LU 6.2 are that it:

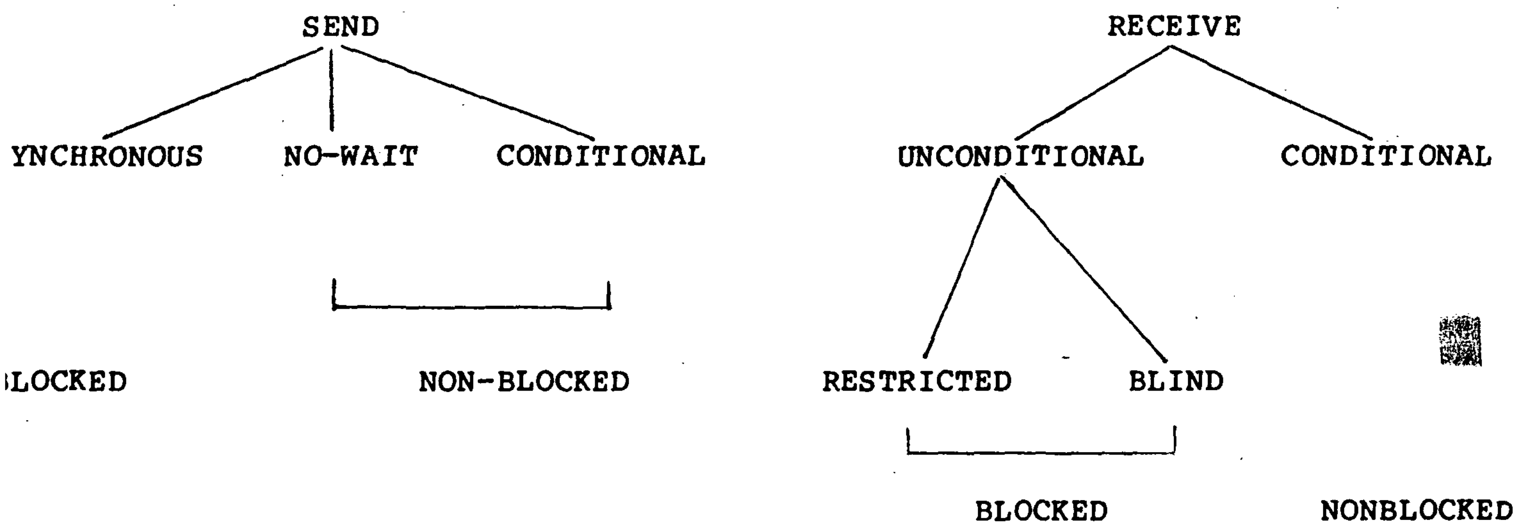


FIGURE 4.2

COMMUNICATIONS PRIMITIVES

- permits multiple applications to communicate through the same process,
- is symmetric,
- can transmit arbitrary binary data,
- employs a half-duplex protocol,
- is designed with the intention to accommodate distributed data base two phase transaction protocols. See Section 5 on Concurrency and Synchronization.

4.3 Data Migration

The Network Operating System must support the remote access of information. Since heterogenous nodes are involved, data types must be preserved across the network between the nodes. Data and character translations and even data restructuring may be supported.

4.3.1 File Service

The File Service provides the capability to the user of performing operations on:

- o the contents of a file,
- o on entire file, or
- o groups of files.

The key elements of a File Service are:

- o file structures supported,
- o specific operations that can be performed on contents of a file, the file, and groups of files,
- o degree of sharing,
- o access control.

The file service is probably the most common service provided by Network Operating Systems. The file service has evolved from its rudimentary form of only transferring entire files to remotely accessing files through commands sent over the network. A representative set of commands is enumerated in Figure 4.3.

The structures of the file will drive the efficiency with which various users will be able to access files and thus establish the suitabiilty of the file structure for various applications. The structure of the file can be viewed in terms of:

- o access structure describing the organization of data contained with a file as it affects a file's contents
  - flat or hierarchical
  - sequential, random, indexed sequential
- o presentation structure describing the syntax of the data items in the file, e.g., are all files treated as binary files or are specific structures supported such as graphics files, etc.

Another major characteristic of the file service is the degree of sharing of programs and data stored in these files and the commands users are permitted to execute. For example, files can be shared at the following levels:

- o entire file,
- o records,
- o field.

When files are accessed users may be permitted to perform:

- o read only operations,
- o write only, modify only operations,
- o read and write operations.

Another major issue is maintaining concurrency of files when

F-CONNECT - file service initiation;
F-RELEASE - file service termination (orderly);
F-ABORT - file service termination (abrupt);
F-SELECT - file selection;
F-DESELECT - file deselection;
F-CREATE - file creation;
F-DELETE - file removal;
F-READ-ATTRIB - file identity attribute inspection;
F-CHANGE-ATTRIB - file identity attribute modification
F-OPEN - file access initiation;
F-LOCATE - FADU location;
F-ERASE - FADU removal;
F-RECOVER - file access recovery;
F-CLOSE - file access termination;
F-WRITE - data storage initiation;
F-READ - data retrieval initiation;
F-DATA - data retrieval initiation;
F-DATA - data transfer
F-DATA-END - data transfer completion
F-TRANSFER-END - data transfer commitment;
E-CANCEL - data transfer cancellation;
F-CHECK - data transfer checkpoint;
F-RESTART - data transfer resumption

FIGURE 4.3

Typical File Service Commands

multiple users are accessing the file. This requires a file locking capability to prevent multiple, concurrent modify accesses. Furthermore, these files may be replicated at different locations, complicating the concurrency control problem.

The relevant standards for the file service is the ISO File Transfer, Access, and Management protocol (IS 8571). To provide compatibility between different local file systems, this protocol employs a virtual file store. A virtual filestore is a data structure representation of the file and corresponds to the file structure perceived by the remote consumer. The local server must perform the translation from the actual file structure to the virtual file structure. This facilitates compatibility for exchange of files among multiple users. For example, if N users are exchanging information, each may have their own file structure and must perform translation to the virtual filestore. Thus N conversions would be required rather than N x (N-1) conversions if each user had to perform conversion to every other user's local structure.

The National Bureau of Standards enhanced the ISO FTAM protocol with three party transfer model. The three party model enables a user to initiate a file transfer and then proceed to another task while the file transfer is performed. Furthermore, the user may be resident on system A and request a file transfer from System B to System C. Traditional file transfer services provide a transfer from only a remote host to the local user system while the local user maintains a connection with the remote file server.

4.3.2 Data Representation Service

The Data Representation Service enables the user to encode data in different syntaxes. The major issues associated with the presentation service are syntaxes supported and the capability to switch syntaxes. The selection of syntax may allow for representation of:

- o binary data,
- o character text,
- o graphics,
- o images.

Characters may include letters, integers, reals, punctuations, and mosaics. Character text may be unformatted such as an ASCII character string or formatted such as time or data representation. Characters may include letters, numbers, punctuations, or mosaics.

The other aspect of the data representation service is if the capability is not provided, then the syntax to be employed between communicating parties must be prearranged. Alternatively the syntax may be negotiated during the setup of a connection. The most sophisticated capability is to switch upon demand during the correction. The ISO Presentation protocol (DP 8822) provides the capability to switch context upon demand.

There are no specific syntaxes embedded in the ISO presentation protocol. Rather the specific encoding can be incorporated as a table. Applicable encoding standards are the ISO Abstract Syntax Notation (DP 8824) and the CCITT X.409 encoding standards for message handling. These are very similar, but do not provide the capability to represent graphics.

4.4 Network Control

The Network Operating System shall provide the means for allocating resources and for controlling the interaction between the network and its nodes.

4.4.1 Network Management Service

The Network Management Service ensures the on-going fault free operation of the network including:

- o network configuration,
- o gathering and processing statistical information,
- o fault detection, isolation, and restoration.

The Network Configuration activity includes the adding or deleting of resources which may include hosts, workstations, as well as server processes. For example a server process to perform specific types of calculations or data base retrievals is viewed as a resource.

The gathering and processing of statistical information has widespread users. It may be for accounting, network signing, or fault detection. Statistics such as connect time, number of connections, lost messages, delays may be gathered and processed.

It is very desirable goal that the NOS perform fault detection, isolation, and restoration service to ensure the error free delivery and maintenance of data in an environment characterized by potential device or network failures.

Features that are incorported into NOS to address these problems are on-line diagnostics and data integrity measures such as periodic file updates, checkpoints, transaction updates, and shadow file servers. These may be either NOS or local operating systems responsibility.

On-line diagnostics ensure that the network is functionally operational at all levels. Typically, this requires a Master Diagnostics Process which periodically communicates with local diagnostics process in all other nodes. The Master Diagnostics Process in responsible for collecting (either by polling or reading a shared file) individual nodal diagnostic statistics and interpet them to isolate abnormal incidence e.g., on high error/retry rate for traffic originating from destined to a particular mode. Alternatively this function could be at least partially distributed.

The data integrity measures ensure that data is not lost or the loss is minimal when a failure occurs. Specific techniques are:

- o Periodic updates to the disk file structure (file control block) to minimize loss of data due to failure,
- o Checkpoints fully exploit the above capability and requires the application/user process to be capable of being suspended while the failure is being fixed and then performs a back-up restart from the most recent update.
- o Transaction update is a combination of the twoapproaches described above. The file server provides the capability to accumulate several file operations (writes/updates) without actually performing them on the disk. The application can indicate when a new transaction begins and let the server accumulate the writes/updates until the server is informed that the transaction is complete. At this time the server performs the actual disk updates in such a manner that if there is a failure before all updates have been performed the file will behave as if no changes were made such that a transaction becomes an atomic operation.
- o Shadow file server: Provides a second file server with the same network name and address as the primary file server to co-exist on the LAN: it performs all the same disk writes as the primary (active) file server but does

not respond to reads, however it does maintain control information (such as current read printer) necessary to respond to a read. If and when the primary server fails, it immediately takes over and starts responding to reads.

The key issues associated with during so are:

- o degree to which recovery feature should be built into the NOS versus the application,
- o degree to which recovery features should be built into local hardware.
- o use of automatic versus manual reconfiguration.

This is closely related to maintaining concurrency of user and directory files. In particular there is a movement within the OSI environment to restructure layer 7, the application layer, to provide the recovery service. A draft proposal exists for implementing it, namely the Commitment, Concurrency, and Recovery Service (DP 8650) as a layer 7 protocol.

4.4.3 Directory Service

The Directory Service provides the capability to the user of locating other users, and resources (hosts) or services in the network. Note the file directory is included in the file service not in the Directory Service. This separation is effected because user and resource directories are updated infrequently relative to updates in the file directories. Thus different approaches are applicable.

Salient features of the the Directory Service for users and resources are:

- o naming and addressing convention,
- o directory architecture (centralized, distributed, hybrid),
- o directory size and breadth limitations,

- o directory access methods,
- o strategy for establishing and updating the directory.

The establishment of a naming and addressing convention requires that resources and services be uniquely identified by name in the directory such that the requisite network address information can be maintained. The major issue associated with addressing is maintaining currency of the address associated with new users or resources and modifications associated with "Named" users whose address may change.

The network directory architecture can affect cost and performance issues. For example, whether the storage of directory information is centralized or distributed will affect response time associated with retrieving directory information, availability of directory information, and complexity of maintaining current information. The size of the directory (number of entries) and breadth of the directory (number of users and resources) will be, to a large extent, driven by the directory architecture.

There are two primary techniques for accessing the directory service to obtain address information. A user may either:

- o query a specific directory to determine if that directory contains the specific information,
- o send a broadcast query to all directories requesting the address information, but only the directory with the required information will respond.

Furthermore when the directory becomes distributed, maintenance of the directory becomes a distributed data base problem and significantly complicates maintaining currency.

Although there is work currently in process for the development of a directory standards, such as standard has not been forthcoming.

4.5 Common

Common services consist of services provided to any two or more of the services described above.

4.5.1 **Transport Services** (Layers 1-5 of OSI Reference Model)

The Transport Service provides for the logical multiplexing of connections over a single transmission media and the delivery of information from source process (device) to destination process (device). Depending upon the environment this may include OSI protocol layers one to five or some subset of these layers. The data should be reliably delivered in the sequence that the data was entered for most services and applications employing the transport service.

The scope of the transport service depends on the local environment. If the local environment is a local area network, then only physical, data link and session layer protocols may be required. If multiple local area networks of the same kind are employed, then a simple network layer protocol would also be required and a bridge device employed to concentrate the multiple local area networks. If different types of local area network or a wide area and local area network are concatenated then an Internetworking protocol would be required. This would provide such additional functionality as packet fragmentation (to allow networks with different maximum packet sizes to be linked) and speed conversion.

The techniques employed by layers 1-5 to provide the transport layer service should be transparent to the user services and applications. For example the use of connection based protocols versus datagrams as well as the use of virtual circuits is germane only to layers 1 to 5.

4.5.2 **Network Security Service**

The Network Security Service ensures that only the users/processes with the requisite authorization level are permitted to use other network services. This includes:

- o authorization to read or read and modify files,
- o authorization to communicate only among a specified user group,
- o authorization to invoke user services, e.g., file service, message handling service, job transfer, directory service,
- o authorization to invoke system services such as network management.

The use of authorization lists and user passwords is the common technique for implementing the security service.

5.0 SURVEY RESULTS

In this section CTA summarizes the NOS survey results. First the key trends in NOS development are summarized. Then individual summaries of relevant vendor products and research efforts are presented. Each of these narratives is supplemented by a filled-in product/project template in the Appendix. These templates provide concise single page summaries of the products/projects CTA evaluated.

5.1 Key Trends and Observations

Most NOS's were being developed for environments employing UNIX or IBM PC DOS as the local operating system. Of the 28 vendors surveyed, 10 provided UNIX based systems while 16 provided PC DOS based systems. Some vendors were employing their own proprietary operating systems such as Intel using its iRMX operating system and DEC using various operating systems supporting DECnet. However, Intel has formulated its Open Net architecture which provides interoperability with a UNIX look-alike, XENIX, and with PC DOS. Also, DEC has plans to offer DECnet under UNIX, and Technology Concepts intends to offer a plugin board for the IBM PC XT and AT supporting DECnet.

The most commonly offered service is a file service because of its historical significance. In many early developments, a NOS was used to provide file access to workstations without local disks. Although many workstations now have their own local disk, the demand for information is insatiable and users still need access to remote files. Similarly, there was an important requirement to share printers; thus print service evolved. This is still an important service.

These services then evolved to user-to-user communication providing electronic mail (store and forward messaging) and program-to-program communication using the remote procedure call. All of these services are commonly available in NOS products.

Historically, the developments associated with UNIX have preceded the PC DOS developments. Developers working with UNIX have tended to integrate the NOS with the UNIX Kernel and reducing its portability. Thus even though UNIX is available on many hosts, porting the NOS to a new UNIX host is not trivial. The PC DOS developers have tended to build a shell around the PC DOS, which theoretically makes it more portable. However, PC DOS is limited to PC class hosts.

Host support is largely in three main areas:

- o DEC VAX in the minicomputer class,
- o IBM PC in the personal computer class,
- o Motorola 68000 based computers in the workstation.

Many vendors have developed their own custom hosts based on the 68000 such as SUN MICROSYSTEMS and Charles River Data Systems.

Support of the OSI model is minimal. Early LAN developers focused on achieving performance and in doing so, implemented simpler protocols with fewer protocol layers. Vendors who do adhere (or claim to adhere) to the OSI model typically do not support all layers.

Only recently have vendors tended to provide an open system to communicate with systems provided by other vendors. With the increasing acceptance of the OSI Reference Model and associated protocols interoperability among vendors will become more common.

Intel, with its Open Net architecture and Charles River Data Systems are leaders in this area. Intel supports file service interoperability among PC DOS, XENIX, and iRMX hosts using OSI protocols up to layer 4 and protocol conversions in layers 5 to 7. Charles River Data Systems is developing OSI protocols for all layers.

One of the most significant steps in the acceptance of OSI and associated protocols is the promulgation of the Manufacturing Automation Protocol by General Motors. Although details of the specific protocols to be employed are yet to be determined, it is clear that it will be OSI based. Thus, a significant part of industrial automation applications will employ OSI.

5.2 Individual Vendor/Source Summaries

This section provides, in narrative form, a description of the salient characteristics of the relevant products and research efforts identified by CTA's survey. For consistency with Section 4 of this report, each product or project address the NOS functional service requirements listed below:

- User Communication
  - Message Handling
  - Print
  - Terminal Handling
  - Conferencing
- Job Migration
  - Remote Procedure Call
  - Job Transfer and Manipulation
  - Interprocess Communication
- Data Migration
  - File Service
  - Data Representation

Network Control
- Recovery Service
- Network Management
- Directory

Common
- Transport
- Security

The narrative is supplemented by the product templates provided in Appendix A to this report.

Sections 5.2.1 through 5.2.13 present the narratives for 13 NOS products surveyed by CTA. These products were selected since collectively they represent implementations of all NOS functional characteristics identified in Section 4. Also, each product described offered a unique service(s) and/or design feature(s) in its implementation which warranted description. CTA has provided narratives for only a few of the many commercial and research products identified during this survey. This was done for two reasons. First, a greater level of detail is provided on each of the 12 products than could otherwise be supplied. Second, CTA recognizes that, ultimately, this survey would be used by NASA to produce a build for an NOS. This also caused CTA to focus principally on detailed descriptions of commercial products only.

The first product, LOCUS (5.2.1) offers a network-wide, replicated, distributed file system which may be accessed transparently by any user on the network. This transparency is extended to all devices as well as permiting remote access by all users in a heterogeneous environment. LOCUS stands out among the commercial products surveyed since it offers the user a comprehensive set of NOS features in its implementation. All seven layers of the ISO/OSI model are supported and described in the next product, UniverseNet (5.2.2) by Charles River Data Systems (CRDS). CRDS's product is based on a real-time, proprietary UNIX-based O/S and offers efficient, fast networking in a homogeneous environment of CRDS workstations. The real-time performance is important in the NASA environment. Section 5.2.3 describes the Sun Microsystem's product NFS 2.0. Key features of NFS are support of a heterogeneous UNIX-based environment and, most importantly, standards proposed for remote procedure call and data representation services. The Newcastle Connection (5.2.4) available from Portable Software, Inc. is another UNIX-based product. Its prominent feature is the implementation of networking capabilities as a set of software residing between the kernel and the shell. This implementation facilitates ease of portability of the product since no modifications have been made to the operating system. Many vendors have used or based their networking product on UNIX, and specifically, the Berkeley Software Distribution (BSD). Section 5.2.5 focuses on the capabilities of BSD version 4.2 Its salient features extend the basic UNIX capabilities with interprocess communication, virtual memory support, and network file and printing services. Fusion, a product offered by Network Research Corporation, is described in Section 5.2.6. It offers a comprehensive implementation of NOS characteristics and conforms to the OSI Model. No modification to the local O/S kernel is required since it is implemented as a device driver to the O/S code. Fusion supports a wide variety of processors and operating system environments. OpenNet, a product offered by Intel, is described in Section

5.2.7. Its architecture is based upon the OSI model and its salient characteristic is its interoperability with the iRMX, XENIX, and PC-DOS operating systems. This feature emphasizes its portability.

There are many software products for the IBM PC class (and compatibles) that were surveyed by CTA. Most could be categorized as print and/or file servers. Two products which offered the most networking capabilities are Novell's Netware/86 (5.2.8) and Microsoft's Networks 1.0 (5.2.9). The former is a portable product which has achieved support from a number of LAN hardware vendors. The latter is an extension of MS-DOS 3.1 and is important in light of the number of PC's and compatibles which exist in the marketplace today. As such, this product has the potential of becoming a de facto standard for networking IBM PC's. Section 5.2.10 describes Digital Equipment Corporation's product DECnet. It was chosen since it represents an architecture which, although is proprietary, parallels and is a competitor of the ISO/OSI model. DECnet currently supports many of the NOS characteristics. Of particular interest in that (1) it will be available for UNIX-environments, and (2) a Massachusetts company, Technology Concepts, is producing the printed circuit board equivalent for DECnet. Sections 5.2.11 and 5.2.12 focus on the products QNX and Apollo Domain respectively. The former is offered by a Canadian company, Quantum Software Systems, and offers a UNIX-based networking environment for IBM PC's. Apollo Domain's architecture characterizes both a proprietary operating system and networking (token ring LAN) environment for their MC68000-based workstations. Apollo has implemented an object-oriented design for the Domain network. It is of interest since it is representative of a proprietary distributed operating system (DOS) but,as such, was given a lower priority in this survey.

Tables 5-1 through 5-6 summarize, in matrix form, the NOS features supported by the vendors surveyed by CTA. Vendors and their products are listed in alphabetical order and grouped according to the operating system they are based upon: UNIX (or a derivative), PC-DOS/MS-DOS (and extensions), and proprietary implementations. Some vendors have been listed in more than one group since their product has been implemented in more than one O/S environment. It should be emphasized that these charts be utilized to ascertain only whether a vendor has included some level of the corresponding NOS characteristic. The level of implementation for the vendors listed varies widely. Succeeding paragraphs provide a greater level of detail regarding each implementation.

ORIGINAL PAGE IS
OF POOR QUALITY

Vendor Name	Product	UNIX	O/S Base PC or MS DOS	Other
Apollo Computer, Inc	Apollo Domain	*		
Digital Equipment Corp.	DECnet	4 QTR 1985		
Charles River Data Systems	UniverseNet	*		
Intel	OpenNet	*		
Lantech Systems	uNETix	*		
Locus Computing Corp.	LOCUS	*		
Network Research Corp.	FUSION	*		
Portable Software, Inc	The Newcastle Connection	*		
Quantum Software Systems	QNX 2.0	*		
Sun Microsystems	NFS 2.0	*		
Touchstone Software	The Connectables	*		
AST Research	PCnet-II		*	
Corvus Systems	Omninet		*	
Davong Systems	Multilink		*	
Digital Research Inc	DRNet		*	
Intel	OpenNet		*	
Microsoft	MS-NET 1.0		*	
Nestar	Plan 3000,4000		*	
Network Research Corp.	FUSION		*	
Novell	Netware		*	
Orchid Technology	PCnet		*	
Quadram	Quadnet		*	
Televideo	PM/16		*	
3Com	Etherseries		*	
Touchstone Software	The Connectables		*	
Ungermann-Bass	NET/ONE		*	
Xcomp	X-Net		*	
Alcyon	DFS			Regulus
Apollo Computer, Inc	Apollo Domain			AEGIS
Digital Equipment Corp.	DECnet			VMS,RSX
Digital Research Inc	DRNet			CP/M
Intel	OpenNet			iRMX
Multi Solutions, Inc	S1			S1
Network Research Corp.	FUSION			VMS
Network Systems	NETEX			IBM/MVS,DEC/VMS
Sun Microsystems	NFS 2.0			VMS
Touchstone Software	The Connectables			McIntosh
Wang Laboratories	WangNet			(TBS)

NOTE: Some products have multiple entries since they are supported in more than one O/S environment.

TABLE 5.1

trix of Vendor Product Services

Vendor		User Communications			
Name	Product	MHS	Print	TMS	Conf
ollo Computer, Inc	Apollo Domain	EM	*	*	
gital Equipment Corp.	DECnet	TBD when released			
arles River Data Systems	UniverseNet	EM,TD	PS	(1)	
tel	OpenNet	EM	PS		
ntech Systems	uNETix	*	*	VT	
cus Computing Corp.	LOCUS	EM	*	*	
twork Research Corp.	FUSION		*	VT	
rtable Software,Inc	The Newcastle Connection	*	*	*	
antum Software Systems	QNX 2.0	TD	PS	*	
n Microsystems	NFS 2.0				
uchstone Software	The Connectables	EM	PS	VT	
T Research	PCnet-II		*		
rvus Systems	Omninet		*		
vong Systems	Multilink		*		
gital Research Inc	DRNet		*		
tel	OpenNet		*		
crosoft	MS-NET 1.0		*		
star	Plan 3000,4000	*	*	*	
twork Research Corp.	FUSION	*	*	*	
vell	Netware	*	*		
chid Technology	PCnet	*			
adram	Quadnet	*	*		
levideo	PM/16	*	*		
om	Etherseries	EM	PS		
uchstone Software	The Connectables	EM	PS	VT	
germann-Bass	NET/ONE	*	*		
omp	X-Net		*		
cyon	DFS				
ollo Computer, Inc	Apollo Domain	*	*	*	
gital Equipment Corp.	DECnet	EM,TD	*	VT	
gital Research Inc	DRNet		*		
tel	OpenNet		*		
lti Solutions, Inc	S1	EM,TD			
twork Research Corp.	FUSION	*	*	*	
twork Systems	NETEX				
n Microsystems	NFS 2.0				
uchstone Software	The Connectables	EM	PS	VT	
ng Laboratories	WangNet	*		VT	

TE: Some products have multiple entries since they are supported in more than one O/S environment.

- PRINT SERVICE COMMON TO ALL
- MESSAGE HANDLING IN MOST PRODUCTS
- TERMINAL HANDLING PRINCIPALLY IN UNIX & PROPRIETARY SYSTEMS
- CONFERENCING LACKING

TABLE 5-2

(1) Virtual Terminal may be implemented by user applications software

(2) Codes: EM = Electronic Mail
TD = Terminal Dialogue
PS = Print Server
VT = Virtual Terminal
* = service is supported

Matrix of Vendor Product Services

Vendor Name	Vendor Product	Job Migration RPC	Job Migration JT&M	Job Migration IPC
Apollo Computer, Inc	Apollo Domain	*		*
Digital Equipment Corp.	DECnet		TBD when released	
Charles River Data Systems	UniverseNet			*
Intel	OpenNet	*		*
Lantech Systems	uNETix			*
Locus Computing Corp.	LOCUS	*		*
Network Research Corp.	FUSION	*		
Portable Software,Inc	The Newcastle Connection	*		*
Quantum Software Systems	QNX 2.0	*	*	*
Sun Microsystems	NFS 2.0	*		
Touchstone Software	The Connectables			
AST Research	PCnet-II			
Corvus Systems	Omninet			
Davong Systems	Multilink			
Digital Research Inc	DRNet			
Intel	OpenNet			*
Microsoft	MS-NET 1.0			
Nestar	Plan 3000,4000			
Network Research Corp.	FUSION	*		
Novell	Netware			
Orchid Technology	PCnet	*		
Quadram	Quadnet			
Televideo	PM/16			
3Com	Etherseries			
Touchstone Software	The Connectables			
Ungermann-Bass	NET/ONE			
Xcomp	X-Net			
Alcyon	DFS	*		
Apollo Computer, Inc	Apollo Domain	*		*
Digital Equipment Corp.	DECnet	*		*
Digital Research Inc	DRNet			
Intel	OpenNet			*
Multi Solutions, Inc	S1			*
Network Research Corp.	FUSION	*		
Network Systems	NETEX			
Sun Microsystems	NFS 2.0	*		
Touchstone Software	The Connectables			
Wang Laboratories	WangNet	*		*

NOTE: Some products have multiple entries since they are supported in more than one O/S environment.

- JOB MIGRATION LACKING IN DOS SYSTEMS
- DEFFICIENCY IN JT&M OVERALL

TABLE 5-3

Matrix of Vendor Product Services

Vendor Name	Product	Data Migration File Svc	Data Migration Data Rep
Apollo Computer, Inc	Apollo Domain	*	
Digital Equipment Corp.	DECnet	TBD when released	
Charles River Data Systems	UniverseNet	FT,FL,RL	
Intel	OpenNet	FA,SH	
Lantech Systems	uNETix	FT,FA	
Locus Computing Corp.	LOCUS	FT,SH,FR	*
Network Research Corp.	FUSION	FT	*
Portable Software,Inc	The Newcastle Connection	FA	
Quantum Software Systems	QNX 2.0	FT,SH	*
Sun Microsystems	NFS 2.0	*	*
Touchstone Software	The Connectables	*	*
AST Research	PCnet-II	*	
Corvus Systems	Omninet	*	
Davong Systems	Multilink	*	
Digital Research Inc	DRNet	FA,FL,RL	
Intel	OpenNet	FA,SH	
Microsoft	MS-NET 1.0	*	
Nestar	Plan 3000,4000	*	
Network Research Corp.	FUSION	*	*
Novell	Netware	*	
Orchid Technology	PCnet	*	
Quadram	Quadnet	*	
Televideo	PM/16	*	
3Com	Etherseries	FS	
Touchstone Software	The Connectables	*	*
Ungermann-Bass	NET/ONE	*	
Xcomp	X-Net	*	
Alcyon	DFS	FA	
Apollo Computer, Inc	Apollo Domain	*	
Digital Equipment Corp.	DECnet	FT,SH	*
Digital Research Inc	DRNet	FA,FL,RL	
Intel	OpenNet	FA,SH	
Multi Solutions, Inc	S1	FT,SH	
Network Research Corp.	FUSION	*	*
Network Systems	NETEX	FT	*
Sun Microsystems	NFS 2.0	*	*
Touchstone Software	The Connectables	*	*
Wang Laboratories	WangNet	FT	

NOTE: Some products have multiple entries since they are supported in more than one O/S environment.

- FILE SERVICE IS UNIVERSAL
- DATA REP IS LACKING

TABLE 5-4

(1) Codes:
FT = File Transfer
SH = File Sharing
FS = File Server
FL = File Locking
RL = Record locking
FA = File Access
FR = File Replication
* = service is supported

Matrix of Vendor Product Services

Vendor		Network Control		
Name	Product	Recovery	Net Mgmt	Directory
Apollo Computer, Inc	Apollo Domain		*	
Digital Equipment Corp.	DECnet		TBD when released	
Charles River Data Systems	UniverseNet			
Intel	OpenNet			
Lantech Systems	uNETix			
Locus Computing Corp.	LOCUS		*	*
Network Research Corp.	FUSION		NS	
Portable Software,Inc	The Newcastle Connection			
Quantum Software Systems	QNX 2.0			*
Sun Microsystems	NFS 2.0			
Touchstone Software	The Connectables		*	
AST Research	PCnet-II			
Corvus Systems	Omninet			
Davong Systems	Multilink			
Digital Research Inc	DRNet			
Intel	OpenNet			
Microsoft	MS-NET 1.0		*	
Nestar	Plan 3000,4000			
Network Research Corp.	FUSION		*	
Novell	Netware			
Orchid Technology	PCnet			
Quadram	Quadnet			
Televideo	PM/16			
3Com	Etherseries			
Touchstone Software	The Connectables		*	
Ungermann-Bass	NET/ONE			
Xcomp	X-Net			
Alcyon	DFS			
Apollo Computer, Inc	Apollo Domain			
Digital Equipment Corp.	DECnet		NS	
Digital Research Inc	DRNet			
Intel	OpenNet			
Multi Solutions, Inc	S1			
Network Research Corp.	FUSION		*	
Network Systems	NETEX			
Sun Microsystems	NFS 2.0			
Touchstone Software	The Connectables		*	
Wang Laboratories	WangNet			

NOTE: Some products have multiple entries since they are supported in more than one O/S environment.

- RECOVERY PROC. LACKING
- MINIMAL NET. MANAGEMENT

(1) Codes: NS = Network Statistics

TABLE 5-5

Matrix of Vendor Product Services

Vendor Name	Product	Common Transport	Security
Apollo Computer, Inc	Apollo Domain	*	*
Digital Equipment Corp.	DECnet	TBD when released	
Charles River Data Systems	UniverseNet	*	*
Intel	OpenNet	*	*
Lantech Systems	uNETix	*	
Locus Computing Corp.	LOCUS	*	*
Network Research Corp.	FUSION	*	
Portable Software, Inc	The Newcastle Connection	*	LP
Quantum Software Systems	QNX 2.0	*	OS
Sun Microsystems	NFS 2.0	*	*
Touchstone Software	The Connectables	*	*
AST Research	PCnet-II	*	*
Corvus Systems	Omninet	*	*
Davong Systems	Multilink	*	
Digital Research Inc	DRNet	*	FP,OS
Intel	OpenNet	*	*
Microsoft	MS-NET 1.0	*	*
Nestar	Plan 3000,4000	*	
Network Research Corp.	FUSION	*	
Novell	Netware	*	*
Orchid Technology	PCnet	*	*
Quadram	Quadnet	*	*
Televideo	PM/16	*	*
3Com	Etherseries	*	FP
Touchstone Software	The Connectables	*	*
Ungermann-Bass	NET/ONE	*	
Xcomp	X-Net	*	
Alcyon	DFS		
Apollo Computer, Inc	Apollo Domain	*	
Digital Equipment Corp.	DECnet	*	LP,FP,OS
Digital Research Inc	DRNet	*	FP,OS
Intel	OpenNet	*	*
Multi Solutions, Inc	S1		
Network Research Corp.	FUSION	*	
Network Systems	NETEX	*	*
Sun Microsystems	NFS 2.0	*	*
Touchstone Software	The Connectables	*	*
Wang Laboratories	WangNet	*	

NOTE: Some products have multiple entries since they are supported in more than one O/S environment.

- SECURITY:
  - USUALLY PASSWORDS
  - LOGON
  - FILE & RECORD ACCESS CONTROL

(1) Codes: LP = Logon Password
FP = File Password
OS = "Other" Security (e.g., private volumes)
* = service is supported

TABLE 5-6

5.2.1 LOCUS

Locus is a distributed operating system (DOS) produced by the LOCUS Computing Corporation of Santa Monica, CA. This UNIX-based product represents an outgrowth of research originally performed at UCLA (See "The LOCUS Distributed Operating System" by G. Popek et al, ACM June 1983). Its derivation from UNIX includes modifications of its internals and the incorporation of considerable extensions. LOCUS operates on an Ethernet network utilizing a propriety data transfer protocol. A heterogeneous computer environment consisting of DEC VAX and Motorola M68000 CPU's is supported.

Salient features of the LOCUS DOS include:

- o its distributed file system,
- o remote tasking,
- o dynamic reconfiguration,
- o heterogeneous computer environments, and
- o system management.

LOCUS supports: transparent access to data through a network-wide file system (a superset of the UNIX tree structure naming hierarchy); automatic replication of storage; and transparent distributed process execution.

As a distributed version of UNIX, LOCUS is object code compatible with Berkeley UNIX V4.1 and V4.2, and UNIX System V. The ISO/OSI seven layer model is not implemented in LOCUS' design. This decision appears to have been made in order to provide users maximum product performance. Because the protocols have been simplified, LOCUS minimizes the protocol and message processing overhead by performing much of the processing at interrupt level rather than in a user process. Telephone conversations with the vendor emphasized that due to the extensive efforts to develop the existing product, the ISO/OSI Model was not likely to be implemented in the near future. Nevertheless, LOCUS provides valuable insight to state-of-the-art NOS technology.

Transparency is the key thrust of the LOCUS DOS. This is accomplished by making a collection of machines appear to the user as a single processor with a root file system. Programs can run on any machine in the system. A unique pathname is associated with each file in the system. This lends itself to data and name transparency allowing uniform access to data in the system regardless of location. Coupled with the transparency philosophy is LOCUS' design feature of file replication. This significantly increases the system reliability and data availability. LOCUS goes to great pains to assure the user that in the event of a file update failure, the file is left in either its old or new state and not something inbetween. Updated files are copied as appropriate throughout the system automatically to assure consisting of replicated files throughout the network. Information maintained at nodes allows the system to automatically select the latest version of a file. Replication of files is dynamic and the degree of it is under direct user control. To assure pathnames to all files (remote or local) are still accessible to the user in the event of failure, LOCUS extends it replication philosophy to the entire directory as well.

Transparency is extended to devices in general allowing, for example, remote computer access from a user's terminal, remote access to line printers, and host access of remote disks. The installation does not rely on the need for a dedicated file server in the network, although, any machine may be so designated if some benefit may be derived from doing this. Remote accesses are performed efficiently in LOCUS primarily due to the networking operations being firmly embedded within the operating system kernel. As a result, all messages take place between local and remote operating systems. Overhead is thus reduced to an absolute minimum and the actions are invisible to user programs.

LOCUS' distributed file system is its dominant architecture feature. Four essential elements of the file system are its distributed naming, catalog, synchronization facilities, integrated replicated storage supports, and the mechanisms to allow partitioned operation. The file system is a functional superset of the UNIX tree structure naming system. LOCUS has extended this in three areas. First, all file system objects on all machines are represented by the single LOCUS tree structure. Since LOCUS names are fully transparent, it is not possible to determine the location of a resource within the network from its name. Second, as noted earlier, LOCUS files are replicated. It is the system's responsiblity to keep all copies current and assure that access requests are provided by the most recent version. Third, LOCUS provides a "multiple reader, single writer" synchronization scheme.

The LOCUS approach to support remote service was to build a fast but limited process facility called server processes. Network requests are enqueued for processing by each server process which in turn serially dequeues each request. This design permits efficient network request processing, keeping protocol overhead low. LOCUS' design also includes a "filter" to discern whether a request is for the local or a remote site. This concept has also been implemented by Microsoft in the form of a "redirector". A description of this implementation is provided in Section 5.2.9 - MS-NET 1.0. Local site requests bypass message processing logic usually performed for remote site access and instead, are handled immediately the the local system call processor. This philosophy is in tune with LOCUS' emphasis on the importance of system performance.

Atomic file commit is an important concept in LOCUS. Essentially, this means that all file changes are handled atomically. No changes to a file are permanent until a commit operation is performed. Commit and abort (to undo changes back to previous commit) system calls are supplied. Closing a file commits it. Updates are done using a "shadow page" mechanism which keeps copies of pages in the file which have been updated/changed. These shadow pages are marked in order to track the new/old information. At commit time, the old pages are replaced by the new ones. To abort a set of changes, pages on disk remain on disk and shadowed disk page space is deallocated for reuse. Files may also be shared in LOCUS and a synchronization mechanism for doing so in a networked environment using normal UNIX semantics is supported.

LOCUS supports interprocess communication with the network equivalent of UNIX local pipes (i.e., one way communications between programs). Names and unnamed pipes are supported transparent to the user and are implemented as FIFO circular queues. Intertask communications is accomplished by writing to and reading from pipes. If a reader task is asleep (i.e., suspended) on an empty pipe, it is awakened (i.e., put into active state) when a message is written to the pipe.

Remote tasking is also supported by LOCUS. The major process control functions have been extended from UNIX. The UNIX "fork" call creates a new process running the same program as the caller. The UNIX "exec" call replaces the code and date of a running process with a new program and data image. LOCUS utilizes forks to cause a new process to be created locally or remotely. Exec has been extended to allow a process to migrate to another site as it replaces its image. "Run" is a new (LOCUS) system call which creates a new process and invokes a new program from it immediately. This may be done locally or remotely. Finally, to permit a process to change its execution site, while in the midst of execution, LOCUS has added a fourth call; "migrate". The user can influence and/or control process execution sites. Default execution is the user's local site. Via the "setxsites" system call, the user provides specifications required of a site for process execution, a list of sites to be tried in order, or a list of site types (e.g., DEC/VAX, M6800, IBM/PC, etcetera).

The transparency of LOCUS has also been applied to network topology. The system strives to insulate users from network reconfigurations. Key to this is the assumption that the network is fully-connected (the underlying LOCUS protocols assume this!).

Heterogenous computer environments are also handled transparently by LOCUS. LOCUS may be ported to other machines in a manner similar to porting UNIX. It is written in a strongly typed version of the "C" programming language to minimize machine dependencies. LOCUS handles byte recording (if necessary) for different word (2 byte) and long word (4 byte) representations; ASCII/EBCDIC and similar data translations; contains functions in its kernel to transform incoming network message formats.

LOCUS also supports a product, PC-Bridge, which enables PC users to communicate with mainframe hosts. At present, only AT&T 3B-series host processors are supported. Hosts must be equipped with either the UNIX or LOCUS distributed UNIX operating system. Personal computers must run in a PC-DOS or MS-DOS environment under Version 2 or subsequent releases. Utilizing an Omninet, Ethernet, or RS-232 facilities, PC-Bridge enables the PC to utilize the mainframe as a file server. There are four software elements which comprise this package.

1. Bridge - Runs on PC - it logs-in and connects user to mainframe host.
2. Bridge Server - Runs on host - it translates and performs I/O requests from the PC.
3. Bridge Handler - Runs on PC - it redirects all system calls to MS-DOS on the host as appropriate.
4. Terminal Emulator - Emulates terminals connected to the host.

This section has attempted to describe the salient characteristics of the LOCUS architecture. Virtually all of the NOS functional service characteristics identified in Section 4 are supported by LOCUS. Although it does not conform to the ISO/OSI model, the LOCUS designers have made the conscious decision to trade this layered approach for system performance considerations. The reader is referred to "The LOCUS Distributed System Architecture" manual (Edition 3.1, June 1984) which provides an excellent description of the LOCUS internals.

5.2.2 **UniverseNet**

Charles River Data Systems (CRDS) of Framingham, MA sells a networking product, UniverseNet, which implements all seven layers of the ISO/OSI model. UniverseNet applications layer software invokes UN/System V or UNOS, both UNIX compatibles, to provide operating system support and implement all other lower layers of the model. Figure 5-1 illustrates the UniverseNet architecture with respect to the layered OSI model and the O/S kernel. Computers (i.e., nodes) communicate via the Ethernet protocol. The CRDS Universe 68 Super microcomputers from a homogeneous (MC68000-based) networking environment for users running UniverseNet. Computers from other vendors which have implemented OSI communications protocols may communicate with those on the UniverseNet via a LAN. UniverseNet has run under the auspices of NBS test and has approved standard implementations for protocol levels 1-L4 including Class 4 Transport protocols. In keeping with the OSI philosophy, CRDS provides network interfaces to the following protocols: TCP/IP,

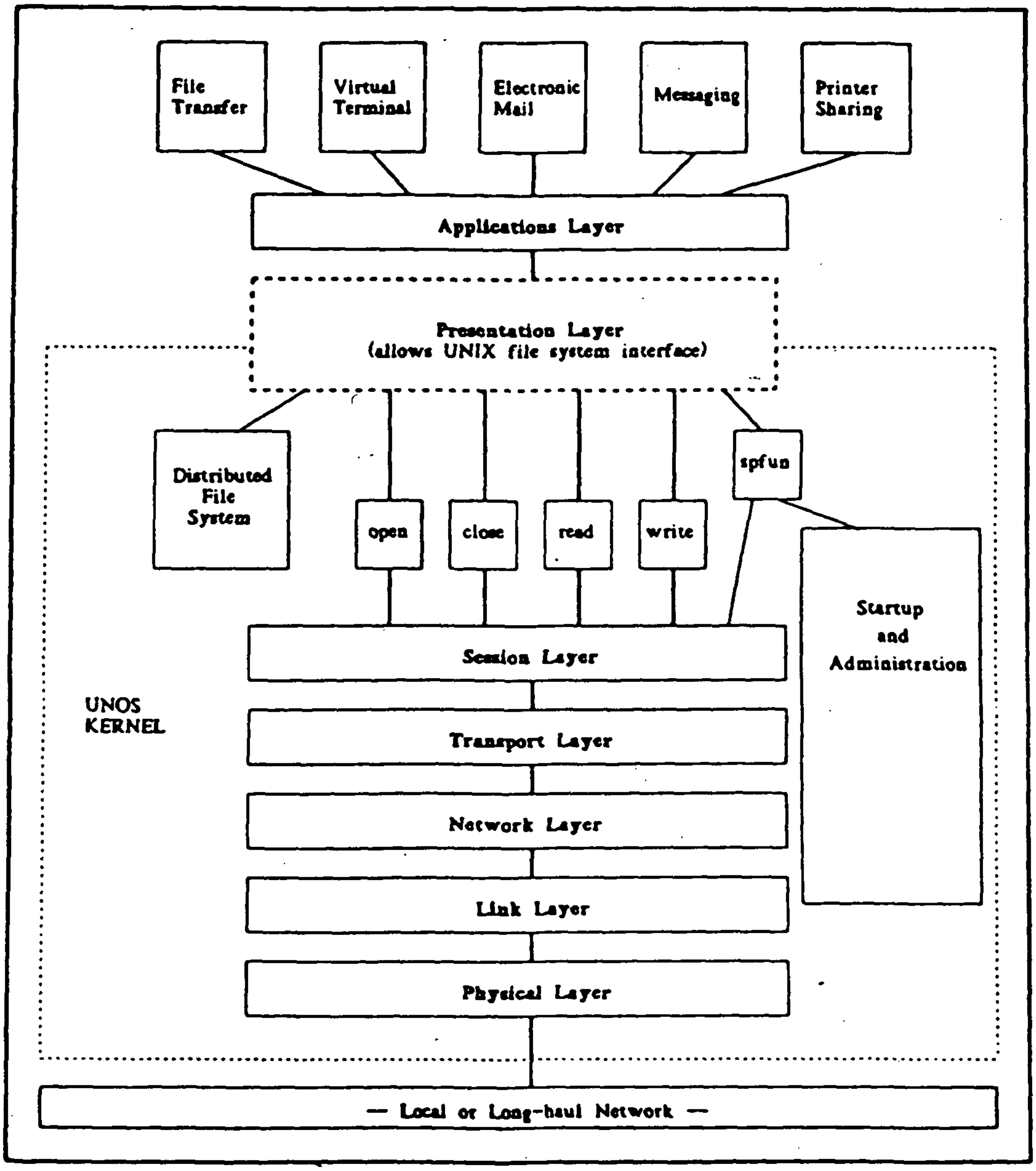


FIGURE 5-1

UniverseNet Layers in the UNOS Environment

Table 5-1

FEATURES	UNOS REV4	UNIX V6	UNIX V7	UNIX V3	UNIX V5
I/O redirection	yes	yes	yes	yes	yes
File independent I/O	yes	yes	yes	yes	yes
Pipe process management	yes	yes	yes	yes	yes
Multi-server queues	yes	no	no	no	no
Named pipes	yes	no	no	yes	yes
Signal/alarm facility	yes	ltd.	yes	yes	yes
Process synchronization	yes	no	no	no	ltd.
Multiplexed files	no	no	yes	no	no
Keyed file facilities	opt.	no	yes	no	no
DBMS facilities	opt.	no	no	no	no
Max. file size (bytes)	2^{31}	2^{24}	2^{30}	2^{30}	2^{30}
File name length (bytes)	30	14	14	14	14
File allocation	Bit map	Linked list	Linked list	Linked list	Linked list
Record/file locking	yes	no	no	no	no
Multi-level directories	yes	yes	yes	yes	yes
Bad block handling	yes	no	no	no	?
Exception handling	yes	no	no	no	no
Symbolic debugger	w/ Macros	ltd.	ltd.	yes	yes
Graphics packages	no	no	yes	yes	?
Text formatting	yes	yes	yes	yes	yes
Typesetting package	opt.	no	yes	yes	yes
Compiler writing aids	opt.	no	yes	yes	yes
Environments	yes	no	yes	yes	yes
Process locking	yes	no	yes	yes	yes
User controlled priority scheduling	yes	no	no	no	no
Heuristic timesharing scheduling	yes	yes	yes	yes	yes
Shared text	yes	yes	yes	yes	yes
Shared data	yes	no	no	no	no
User support	yes	no	no	no	ltd.
Update service	yes	no	no	no	ltd.

XNS, BSC, and SNA. Salient features of UniverseNet are listed below:

- o Maximum efficiency via kernel integration with UN/System V or UNOS kernel.
- o Applications include: electronic mail, messaging, printer sharing, file transfer, and remote command execution.
- o Distributed File System
- o Written in "C"
- o LAN and WAN network support
- o Built-in test facilities

Key to UniverseNet are the UN/System V and UNOS Operating Systems. UN/System V is derived from UNIX System V under license from AT&T. It includes the entire UNIX System V program development tool set. UNOS is a CRDS proprietary UNIX-compatible O/S. Its key feature is the incorporation of real-time extensions not available with UNIX. UN/System V and UNOS share the same device drivers, file formats, and object formats to facilitate a fully transportable environment. Table 5-1 depicts the salient features of these two operating systems as well as other UNIX implementations. Both implementations of these two operating systems are linked to the program code of UniverseNet layers 1-5. The system thus treats the network layers as a device accessible via a standard I/O driver. Network performance is thus maximized via this integration of code since it has direct access to system calls and can share data without passing it from layer to layer. Network access to both programs and the file system is identical to any other device access. Existing standard OPEN, CLOSE, READ, and WRITE system calls can still be used without modification.

As with LOCUS, UniverseNet takes advantage of the UNIX pipe facility. Pipes have two implicit limitations:

- o All processes in a pipe must be initiated by the same parent process, and
- o byte streams in pipes consist of data with no status or identification.

CRDS notes that these limitations restrict a server process since it cannot use pipe input to handle multiple processes or processes that were not connected to it at the beginning of its operation. UNOS provides a queue package for server - user situations, event counts for process synchronization, and a number of extensions for production oriented environments to overcome these pipe limitations.

Event counts are the focal point for scheduling and synchronization in the UNOS kernel and user process. They are dynamically defined resources developed to meet the needs of network synchronization. Event counts may be opened by unrelated processes and used concurrently by many processes. They may be used as counters whose value increases monotonically. This facilitates indications of event transitions as well as keeping a record of the number of occurrences of an event. They may also be used for process synchronization such as access to critical resources. By waiting for one or many events, event counts may also be utilized for inter-process communication. Interprocess communication facilities are built such that the two processes may be local or based on two different machines. This is supported by developing a close relationship between local interprocess communications capabilities and the session control layer (L5). Finally, since event counts are not owned by any single process, they may be monitored independently by other processes which take appropriate actions based upon their value.

An additional UNOS extension is file locking. Any process can lock any region of a file. Processes are ensured of reading, updating, and writing any type of logical record without the risk of race conditions. Processes may also be locked (i.e., memory resident) to assure fast response. Pre-emptive priority scheduling and fast context switching are two additional enhancements supplied by UNOS.

The following paragraphs summarize the UniverseNet Applications layer software supplied by CRDS.

Electronic Mail enables users to transparently create, edit, send, receive, and handle messages. This is done utilizing a 'mailbox' structure integrated with UniverseNet. Certified mail can be sent which sends a verification message to the sender.

Electronic Mail is invoked with the email command and permits users to send, receive, and handle messages. The mailer allows users to perform the following functions:

- o Create and send messages to yourself and to other users,
- o Read and edit messages,
- o Respond to messages,
- o Copy messages to files and printers,
- o Create and use additional mailboxes as an on-line filing system,
- o Create and use distribution lists.

Messaging is invoked with the WRITE command. Users can thus send a message directly to anyone logged on to the network. This provides short immediate notice messages that don't wait for users to check their mailboxes.

Systems in the CRDS UniverseNet can share resources (e.g., printers) for maximum efficiency and economy. Systems can be configured to use any printer in the network for default printing.

The file transfer application provides the basis for a distributed file service in the UNOS operating system kernel. The protocol is a subset of the remote file access protocol defined by ISO.

Remote execution application permits users connected to UniverseNet to send commands to remote systems for execution on that system.

A distributed file system links the files on several machines through presentation control (from the OSI reference model's layer 6). These linked files then appear to be on one machine. Programs can reference files without regard for actual file location.

UniverseNet development for this feature allows users to reference UNOS features (such as event counts) across machines. Once the UniverseNet distributed system links the application with the remote file system, only UniverseNet (and the system administrator) know the actual file location.

UniverseNet allows development of a virtual terminal application that could eliminate cable stringing for attaching terminals to other systems. Such an application would allow terminals to interact with remote systems as if they were physically attached.

5.2.3 Sun Microsystems NFS 2.0

NFS 2.0 is a product scheduled for release in the June 1985 timeframe by Sun Microsystems, Inc. of California. NFS utilizes the Ethernet protocol to enable Sun Workstations, DEC VAX, and IBM PC computers to communicate in primarily UNIX O/S environments. Mt. Xinu, a California based company, recently announced its implementation of NFS on VAX computers which run UNIX 4.2BSD. Non-UNIX operating systems (e.g., VAX/VMS) from other vendors are also supported. It is intended for use in a networked environment of multiple server and client workstations. Salient features of NFS 2.0 are:

- o heterogeneous computer environment support,
- o transparent access to shared network resources,
- o shared access to one original file copy for all users,
- o remote procedure call service (RPC),
- o external data representation package (XDR).

NFS does not conform to the ISO/OSI model standard but, instead, has implemented its own protocols. NFS is based upon the UNIX 4.2BSD implementation (see Section 5.2.5 for a description of 4.2BSD features) and resides in the UNIX kernel. The remainder of this section addresses the three facets of the NFS implementation: the network file system, remote procedure call, and external data representation. At present, since this is a soon-to-be-released product, CTA was unable to obtain information on NFS other than the proposed NFC, RPC, and XDR standards.

NFS permits file system sharing in a heterogeneous network of machines, operating systems, and networks. "Servers" are the machines in the network which supply resources to "client" machines which access the network resources. Users are logged-in at clients which in turn run applications programs. VNODES (a re-implementation of UNIX inodes) supply the interface to a particular type of file system (called virtual file systems; VFS's) such as UNIX or DOS. The NFS interface defines traditional file system operations for reading directories, creating files, reading and writing files, and setting file attributes. A UNIX-style file protection is implemented by making use of RPC authentication mechanisms.

RPC is implemented on top of TCP/IP transport protocols. It is similar to the local procedure call model where the caller places arguments to a procedure; transfers control to the procedures; and eventually receives control back from the procedure with its associated results. In the remote version of the procedure call, NFS sends a call message to the server and waits for a reply message. Once received, the message results are extracted and the caller's execution is resumed. The NFS RPC mechanism is applicable for both local and remote process intercommunication.

The XDR is a library of routines which describes arbitrary data structures in a machine-independent fashion. It also supplies primitives to enable users to write the own XDR routines.

5.2.4 The Newcastle Connection

The Newcastle Connection (TNC) is a product of Advanced Microelectronics, Ltd. . It is available from Portable Software, Inc. in Redwood City, CA. Key features of the Newcastle Connection are:

- o UNIX to UNIX Networking
- o Resides between Kernel and shell
- o Appears to Kernel as an Application Program
- o Appears to Application Program as an unchanged Kernel
- o Independent of Hardware Network Communications
- o Provides a distributed capability which is functionally indistinguishable from a conventional single processor UNIX system
- o No modification of existing Source Code necessary to operating System or user programs
- o Provides means to read/write any file, use any device, execute any command or inspect any directory regardless of which system it belongs to
- o Naming structures for files, devices commands and directories of each component UNIX System and joined together into a single naming structure
- o Operates under UNIX and UNIX look-alikes
- o Directories are available across machines
- o Shell commands and System calls apply unchanged across Inter-Machine Communication

The implementation philosophy of TNC is the addition of a software subsystem to standard UNIX. Then, in contrast to the LOCUS or UniverseNet packages, no modification to the UNIX kernel has been performed. By adding TNC software subsystem on top of the UNIX kernel to host computers running UNIX, and connecting them in a local area ring network, users may communicate, share files, execute remote commands, and in general are provided with a distributed computing capability.

All of the standard UNIX conventions are applicable transparently to the user including:

- o file and device access, naming, and protection,
- o interprocess communication, and
- o I/O redirection.

The implementation is not specific to any particular UNIX but is instead applicable to any UNIX look-alike which provides system call level compatibility with the original.

TNC filters out system calls that have to be redirected to another UNIX system and it accepts system calls that have been received from other UNIX systems on the network. A remote procedure call is the key mechanism for communications between the Connection layers on the UNIX systems. The Connection layers accepts names and, by use of mapping tables, determines whether an object is local or remote. It also selects actual communications paths and manages alternative routes transparent to the user and his program.

5.2.5 **Berkeley UNIX 4.2**

Version 4.2 of the Berkeley Software Distribution of UNIX (UNIX 4.2BSD) enhances the basic UNIX capabilities with:

- o virtual memory support,
- o faster and more flexible file access,
- o network file and printing services,
- o interprocess communication supporting a multi-window interface,
- o foreground-background job control,

- automatic reboot after system crashes, and
- a new symbolic debugger.

UNIX 4.2BSD represents the results of a DARPA funded project to further develop the 3BSD (i.e., third BSD for UNIX) system. These enhancements were begun in 1983. A brief description of 4.2BSD's salient characteristics and its networking enhancement follows.

Virtual memory support allows processes as large as 16 megabytes to be run under 4.2BSD. Two processors supporting this capability are the DEC VAX 11/7xx and Motorola MC68020. Increases in the number of users time sharing the CPU and overall system responsiveness can be realized.

A new file system implementation is the second key enhancement included in 4.2BSD. Better structures and algorithms have improved overall file I/O performance. Poor layout of disk blocks and I/O operations involving a maximum of only 512 bytes on traditional UNIX systems were inadequate for large applications or networking environments. UNIX 4.2BSD takes advantage of disk geometry and rotational latency characteristics to build in improved performance. It places consecutive data blocks and related indexing information in neighboring areas on the disk. By storing directory files on a single track or cylinder of the disk, seek time is substantially reduced, thus decreasing overall file access time. File access has been further reduced by minimizing multiple seek and I/O operations to retrieve large files. This is due to the fact that most 4.2BSD files are stored in 4096 byte blocks versus the traditional UNIX size of 512 bytes per block. A 32KB file in traditional UNIX required 64 distinct I/O operations of 512 byte data transfers. Total read time is on the order of two seconds or more. By taking advantage of the 4.2BSD disk I/O enhancements, this time can be reduced to about one tenth the two second file access time.

Networking features in the UNIX kernel are one of the most prominent enhancements to standard UNIX. UNIX pipes support byte-stream communications between processes on a single machine, however, they do not support message-oriented communications nor communications with processes on other hosts. Both unidirectional datagrams and bidirectional virtual circuit connections are supported in 4.2BSD. The former are unacknowledged and thus potentially unreliable messages whereas the latter are acknowledged and thus reliable messages. Data in the messages are not interpreted by the system. It is the responsibility of the application software to perform data conversion, representation, or transformation as required. Communications in 4.2BSD take place between "sockets". SOCK_DGRAM and SOCK_STREAM define datagram and virtual circuit socket types respectively. Server sockets are created and then named in order for processes to contact and use the service. Names are used whenever message exchange between sockets takes place or to establish a connection. The call

S = SOCKET (DOMAIN, TYPE)

creates the socket and

BIND (S, NAME)

associates an address to the socket.

Datagram-based network services may create sockets to send/receive datagrams using the

SENDTO (S,MSG,TO)

and

RECVFROM (S,MSG,FROM)

system calls respectively: where S is the socket to send or receive the message; MSG is an array of bytes; and TO/FROM is the socket address. Virtual circuits use system calls to establish a socket, bind it to an address, then issue calls to queue connections (using LISTEN) and service connections (using ACCEPT). CONNECT-CONFIRM and CONNECT-REFUSE messages notify users of a successful or unsuccessful enqueuing respectively.

UNIX 4.2BSD is marketed by several vendors, one of which is MT XINU of Berkeley, CA.

5.2.6 Network Research Corporation - FUSION

FUSION is a networking software package developed by Network Research Corporation (NRC) of Los Angeles, CA. It is developed in the "C" programming language and is available for IBM-PC, M68000, VAX, PDP-11, 16032, and 8086 processors. FUSION supports UNIX (4.1BSD, 4.2BSD, V7, System3, System 5, Xenix, Ultrix, and Venix), MS and PC-DOS, and VAX/VMS operating systems.

FUSION has implemented both the TCP/IP and XNS protocols and runs in an Ethernet environment. Ethernet controller cards developed by 3Com, Interlan, Ungerman-Bass, and CMC are among those currently supported.

FUSION implements L1 and L2 of the OSI model by means of Ethernet. The FUSION Socket Manager is responsible for L3 and

L4. It is integrated into the local O/S kernel as a device driver and thus does not require any modifications to the local O/S code. OSI layers 5-7 are represented by FUSION's Remote Execution and Virtual Terminal facilities. The FUSION Socket Manager is the interface between the host CPU and the network controller. It also performs as the interface between applications programs and the networking package.

User communication, job migration, data migration, network control, and common services identified in Section 4 of this report are all supported in varying degrees by FUSION. Some of these features are inherent in the support supplied by the local operating system (e.g., networking facilities offered in UNIX 4.2BSD).

In the domain of User Communication, many features of UNIX 4.2BSD (see Section 5.2.5) facilitiate the message handling service. Terminal handling service is supported by FUSION's cirtual Terminal capability. For example, a personal computer under MS-DOS can function as a terminal to a VAX running UNIX or VMS. When using the Virtual Terminal command, the local computer becomes a standard terminal. The user then functions under control of the remote O/S and thus uses its set of commands.

Job Migration services supported include interprocess communication (4.2BSD support) and remote procedure call. Commands executed remotely may be handled by a processor with a different CPU, different operating system, or different network hardware.

Data migration capabilities offered by file services and data representation services are also supported. A single file or complete directories of files may be transferred between different computers and operating systems. FUSION supplies a utility program which reconciles differences in file formats when transferring, for example, a file between a MS-DOS and UNIX-based system.

FUSION offers various network control capabilities primarily as network management services. Network performance analyses and traffic monitoring are supported by the product. These utilities provide the user with a chart of network activity and calculate network data transfer speeds.

As part of the Transport Services, NRC is currently working on an implementation for the TCP/IP protocol. In addition, they plan to offer a gateway between TCP/IP and XNS.

Users are able to develop and add applications programs to the FUSION library routines. These routines are implemented so as to be fully compatible with the network system calls of UNIX 4.2BSD.

5.2.7 **INTEL OpenNet**

The Intel OpenNet architecture provides file service between hosts running the XENIX, iRMX, and PC-DOS operating systems. This architecture is based on the ISO Open Systems Interconnection Reference Model although not all layers are supported. The architecture consists of the following layers:

- o layers 1 and 2 ETHERNET (TM of XEROX Corporation),
- o layer 3 Null,
- o layer 4 ISO Class 4 Transport protocol,

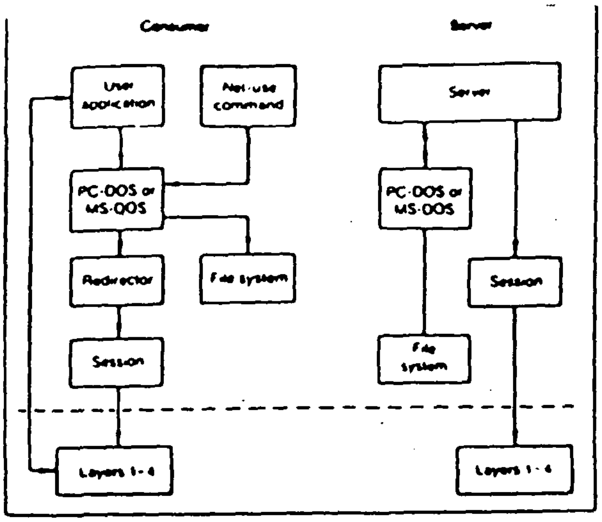
- layer 5 Intel Session protocol,
- layer 6 Null (although Intel claims to have a presentation layer protocol, the functions included in it appear to be applications functions, not data representation functions,
- layer 7 Intel Network File Access protocol.

This architecture is illustrated for each operating system in Figure 5-2. When this architecture is implemented, layers 1 to 4 reside on an outboard communications board while layers 5 to 7 reside on the host. For PC-DOS, layers 1 to 4 execute on an Ungerman-Bass Network Interface Unit card hosted in a PC expansion slot while layers 5-7 execute on the PC native 8088 CPU. For XENIX and iRMX, layers 1-4 reside on the SXM 552 transport and layers 5-7 run under the XENIX or iRMX operating systems in the host. The typical host is an i310 computer which interfaces the SXM 552 over a MULTIBUS (TM of Intel Corporation).

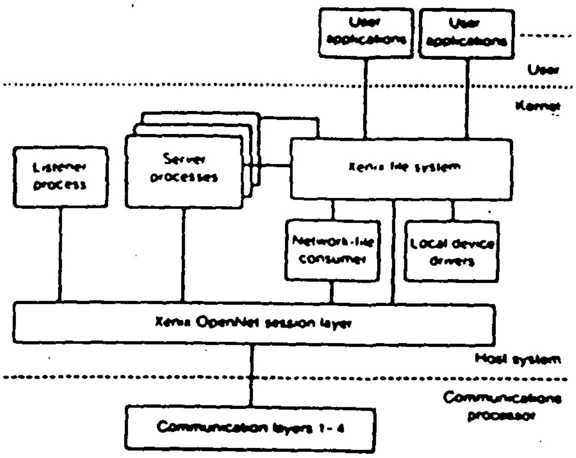
The relatively simply Intel session layer is responsible for mapping symbolic names into network addresses and mapping sessions into transport connections.

The file service enables users to read, write, open, close and manipulate file from a remote location. Thus, a true file access is provided rather than disk sharing or simple file transfer. The salient aspects of the network file access service are:

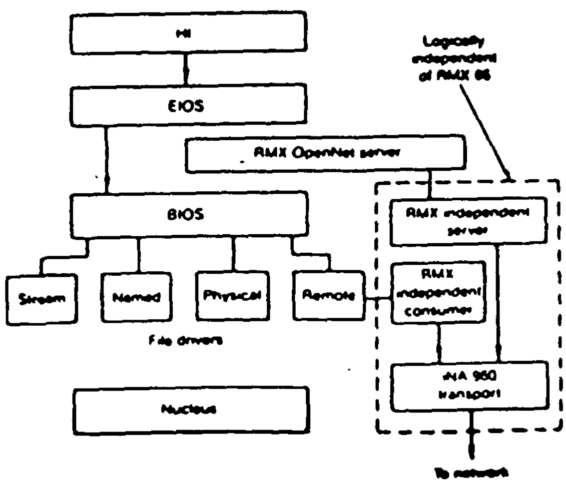
- hierarchical file structure,
- use of pathnames with the name of the system on which the file resides as the first element in the name,
- mapping of server's file protection features to consumer's operating system format,
- record level locking,



MS DOS OPENNET ARCHITECTURE



XENIX OPENNET ARCHITECTURE



iRMX OPENNET ARCHITECTURE

FIGURE 5.2: INTEL OPENNET ARCHITECTURES

rce: "Systems & Software"
March 1985

o file sharing.

A remote print service is also provided for these operating systems.

In addition, selected features of the operating systems will operate across the network when communicating with the same operating system at the remote location. For example, a XENIX system can exchange mail or execute batch jobs. Stream files similar to XENIX pipes can be exchanged using iRMX.

The utility of the OpenNet is in two areas. First, as a retrofit to existing operating systems, it provides networking capability between incompatible systems. However, OpenNet is not a single software module transportable between systems. Second, the XENIX version of OpenNet has the potential to be a portable network operating system. This version is written in "C" and source licenses are available. In addition to running on the Intel 310 Multibus computer, it could be ported to other systems supporting XENIX such as a PC/AT and VAX class machines.

5.2.8 NOVELL NETWARE/86

Netware is a local area network operating system (LANOS) designed for the IBM PC (and compatibles) that can be configured to work with a variety of (over 10) LAN hardware. It does not require a specific PC-LAN board, although Novell does sell its own networking hardware under the name Netware/86.

This section summarizes the Netware/86 product which uses an IBM PC-XT or IBM PC-AT as a central file server. Novell claims to have adapted the Netware/86 software for the following PC-LAN boards:

Corvus	Netware/O	(Omninet)
3Com	Netware/E	(Etherlink)
Proteon	Netware/P	(Pronet)
Davong	Netware/D	
Nestar	Netware/N	
Orchid Tech	Netware/PC	(PCnet)

The PCnet is also sold by AST, Santa Clara Systems and IDE Associates.

Netware is an example of a Dedicated Server. There is one distinct machine which is a dedicated server and other distinct machines which are user machines.

Novell provides its own proprietary multi-tasking Network Operating System that runs on the IBM PC and converts it into a multi-tasking LAN server. This server provides the following functions:

- o File services,
- o Print services,
- o Interprocess Communication (Pipes),
- o Message services,
- o Directory services,
- o Network Management and File Security.

At this time, Netware does not provide:

- o Remote Procedure Call,
- o Job Transfer and Manipulation,
- o Data Representation Services,
- o Terminal handling,
- o Recovery Services.

All Netware/86 networks require the following components:

- **DEDICATED FILE SERVER** - The PC used as the file server may be an IBM PC-XT, IBM PC-AT, IBM PC with a third-party "hardware compatible" hard disk expansion on any one of serveral "IBM PC compatibles" with a hard disk. The maximum disk size currently supported by Netware/86 is 292 MBytes. A 20 MByte File Server requires a minimum of 380 KByte RAM. An additional 1 KByte RAM is needed in the File Server for each additional MByte of disk storage over 20 MBytes.
- **PC Workstations** - with at least 128 KRAM
- **One Network Communication Card** for each PC

A block diagram of a typical Netware/86 configuration is shown in Figure 5-3.

The Netware File Server is a superset of the PC-DOS file system. It supports all PC-DOS file commands. A program can access local disk files (i.e., a disk attached to the user workstation) or network disk files (i.e., files residing in the File Server).

Network files can be opened either for exclusive use of a particular program (private files) or in a non-exclusive mode for concurrent shared usage by several programs. Multiple programs can open a file simultaneously in read-only mode as a private file. Netware provides file-sharing synchronization functions at several different levels:

1. **TRANSACTION UPDATE/PROCESSING:**

   This method allows application software to obtain exclusive control of a set of non-exclusive files when a transaction update is to be performed. The exclusive control is temporary and must be released after each transaction. This technique avoids deadlock conditions that may occur if an application relies on record locking.

2. **FILE/RECORD LOCKING:**

This mechanism allows different applications to modify records of the same file concurrently by giving an application exclusive access to particular record(s) as needed.

An application can specify an entire set of records to the server before actually requesting that the lock be implemented. The server can therefore ensure that all records needed by the application are available before a lock is put on them. This avoids deadlocks between processes which may be working on the same set of records, e.g., two different order processing clerks sharing a common inventory file.

Netware allows application to a lock record(s) for query purposes only. Thus, multiple applications can lock the same records for query. This assures that the selected data is not modified until the lock has been released.

3. **SEMAPHORES**

Netware supports a software library which call on application programs to manage record locking themselves by using general-purpose semaphore mechanisms.

ORIGINAL PAGE IS
OF POOR QUALITY

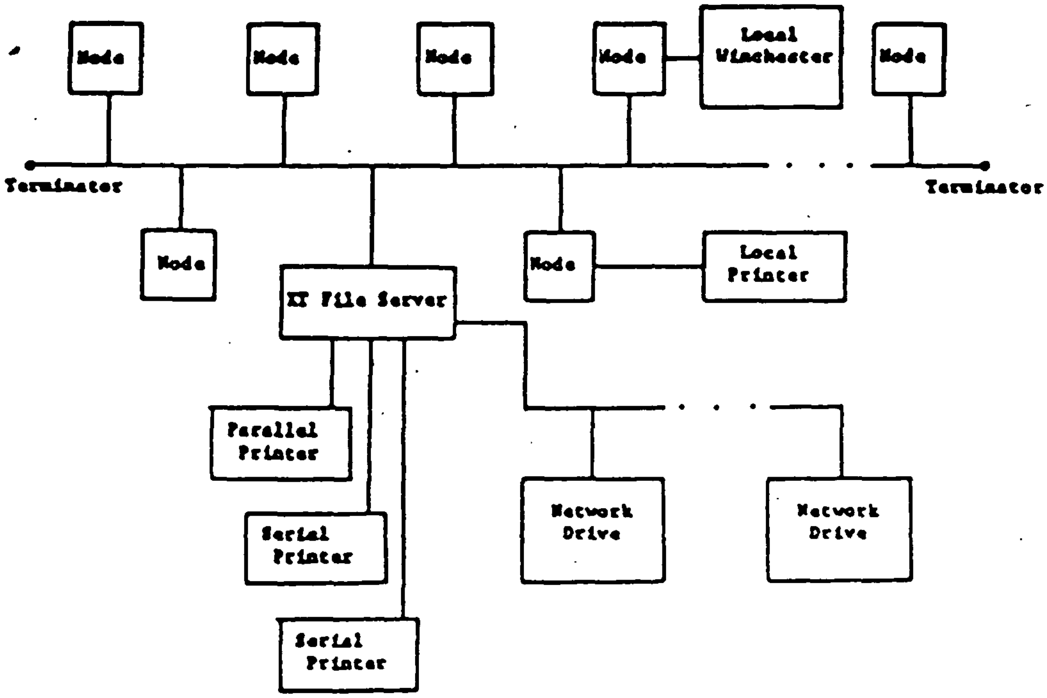


FIGURE 5.3: NETWARE/86 - CONFIGURATION EXAMPLE

Several features maximize the performance of shared file service. Specifically, Novell claims that use of Disk Caching, Directory Caching and Hashing and Elevation seeking results in the highest performance File Server for PC LAN marketplace.

Netware requires a user password for logging in, i.e., accessing the network server for the first time. It further allows users to set passwords within MS-DOS volumes on individual directories as well as individual files. It does not provide security at record or field level. Also, there is only one level of security. If a user has access to a file and then one can perform any operation on it.

Other file services offered by Netware are:

NCOPY: File-to-file copy within the file server without sending the file to the user station

SHOW Commands: To display individual directories and parameters of a particular file such as whether the password has been set.

WHOAMI Commands: Provides the user's network address.

Netware supports a flexible print spooler. It can support up to five independent spooled printers, although the PC-XT is limited to a maximum of three due to hardware restrictions. The spooler allows a user to:

- set forms,
- change queues,
- reroute a print request from one printer to another,
- kill a request before or during printing,
- rewind a print request certain number of pages,
- start and stop a printer,
- spool (print) existing files,

- get printer status.

Netware supports the following messaging services:

- send/receive broadcast messages,
- send/receive personal message,
- send message to system console.

Netware supports a "Message Pipe" for direct communication between two or more processes through the Network Server.

Novell has recently announced its intention to support MS-DOS 3.1. In particular, it claims that the Advanced Netware will be compatible with MS-DOS 3.1 and provide a product superior to the Microsoft MS-Net (described in next section). The product is under development and as such details are not available at this time.

5.2.9 Microsoft/IBM-PC Network

The Microsoft MS-NET is the most recent product introduced to the marketplace by IBM and Microsoft. IBM is the first OEM customer of the software and it has packaged it with Network hardware supplied by Sytek and sells it under the name PC Network (which is different from PC Cluster) First retail shipments of the product were made in May 1985.

Microsoft claims that the following network vendors will be supporting MS-NET:

AST Research

Corvus Systems

Davong Systems

Nestar Systems, Inc.

Proteon, Inc.

Ungermann-Bass

Western Digital

Interlan, Inc.

3COM

Techmar

Orchid

The PC Networking software is being sold as two separate packages.

MS-DOS 3.1 is the most recent extension of MS-DOS which provides support needed for networking and expands the file-name structure to include addressing multiple nodes across the network. IBM's version is called PC-DOS 3.1 and includes certain additional utilities specific to the IBM PC. Both MS-DOS and PC-DOS 3.1 provide an interface (INT 21 hex) to the network redirection program which recognizes requests for I/O as local or remote. It can be viewed as the Client Software.

MS-NET 1.Ø is an extension of MS-DOS 3.1. It provides an operating environment that implements the session layer (layer 5 of the ISO model) and has a file and print server at layer 7. The presentation layer (layer 6) is not implemented (i.e., a null layer). The server runs as an application in a server machine and provides file and print services to the network. In addition, it includes the Redirector and several networking utilities.

The Microsoft Server requires dedicated system resources in the single-tasking MS-DOS operating environment, whereas the IBM PC-Network Program is a non-dedicated server.

In addition, the IBM PC-Network Program includes a Message Service application that enables users to send short messages between workstations.

The IBM PC Network hardware consists of an adapter (in each PC) and a broadband head-end frequency translation unit. The PC-Network adapter supplies processing for five of the seven OSI layers, extending from the physical layer up through the session layer. The NET BIOS handles the interface between the adapter and the host computer (INT 5C HEX). As of May 1985, IBM is the only vendor delivering products based on the MS-DOS 3.1 and MS-NET 1.Ø. It is not clear whether other vendors will use the Microsoft session layer software or build it in the adapter to minimize load on the main processor. MS-NET does not have a NET BIOS, however, it supports the 5C interface. These differences are shown in Figure 5-4.

Since the IBM PC-Network is the only released program at this time, the following description is specific to it.

As indicated, the IBM PC Network Program includes the Server, the Redirector and a Message Service. The server is non-dedicated and as such, can reside in any user station. Each PC (node) on the network must have PC-DOS 3.1 and PC-Net 1.Ø software loaded. However, the individual user can decide, where resources connected to his/her PC can be shared by other users or not. A user must have a local hard disk to allow sharing of devices attached to the particular node. However, a node without a hard disk can access shared devices on other nodes. Also, a diskette drive on a node with hard disk can be shared by other nodes. A maximum of 25 nodes on the network can use the disks,

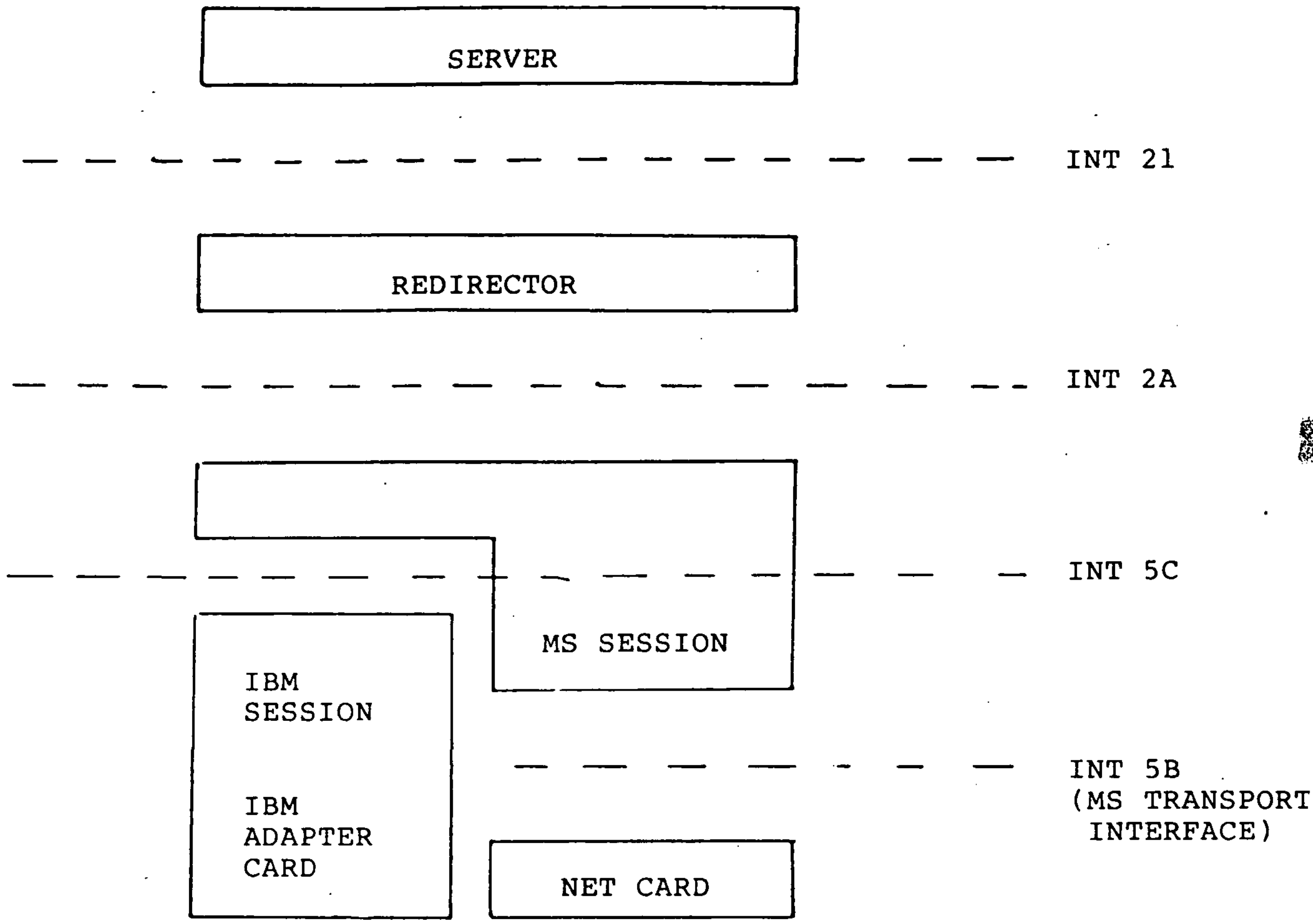


FIGURE 5.4: MS-NET

directories and printers that are permitted for sharing. A particular user can access up to 32 (local or remote) disks, directories and printers. Thus although the PC-Network can connect up to 72 nodes (or up to 1000 nodes using more expensive hardware supplied by SYTEK), a user cannot access all the shared resources on the network at any one time. This is not usually a problem in the PC environment since the PC imposes more severe restrictions due to its limited memory capacity (buffer space) and single-task operating system (PC-DOS).

USER COMMUNICATION: The PC-Network provides the following user communication services:

- Message Handling
  - Send/Receive Messages
  - Pause/Continue Receiving Message
  - Save/View Messages
  - Forward Messages
- Print Services
  - Share a Print Queue
  - Pause/Continue using a Remote Print Queue
  - Stop/Restart a Shared Printer
  - Check/Change Print Queue
  - Change Print Size for a Shared Printer
  - Print a Separate Page (between files)

The PC-Network does not have any facility for Terminal Emulation and Conferencing.

JOB MIGRATION: The PC-Network does not provide any job migration services.

DATA MIGRATION: The PC-Network provides sharing of files or a group of files (contained within a directory) at the file level only i.e., only one user can access (open) a shared data-file at any one time.

The PC-Network does not provide any data-migration services.

NETWORK CONTROL: The PC-Network provides a large set of utilities for Network Management:

- Display names that can receive messages at the user's node,
- Display shared devices at the user's node
- Display remote devices being used by the user
- Pause/Continue using Remote Disk/Directories
- Pause/Continue Sharing devices at the user's station
- Save/Cancel the user's Network Set-up

The PC-Network does not offer any facilities for security and centralized directory services. In fact, it does not seem to provide any method for a user to determine the network configuration (i.e., who are all the users and resources on the network). It seems that a user must have prior knowledge of other users and shared resources on the network, in order to communicate with other users or access remote resources.

The PC-Network is a new product and as such, not much is known about its reliability, performance and other limitations. Also at this time, its functionality is significantly inferior compared to Netware which has been on the market for over two years.

5.2.10 DECnet

DECnet is a family of hardware and software communications products offered by Digital Equipment Corporation (Maynard, Mass). It is currently in Phase IV development and supports many of the functional characteristic features of a NOS identified in

Section 4 of this report. DECnet is based entirely on the Digital Network Architecture (DNA). DNA, a propriety architecture, is similar to the seven layer ISO/OSI architecture model. The relationships between the OSI and DNA layers are illustrated in Figure 5-5. DNA modules resident in a typical DECnet node are illustrated in Figure 5-6. Node-to-node communications protocols are shown in Figure 5-7.

DECnet is designed to run only on the DEC family of computers and operating system environments listed below:

CPU	O/S
VAX	VMS
PDP-11	RSX, RT-11, RSTS
DECsystem-10,-20	TOPS-10, TOPS-20
MicroVAX	MicroVMS, VAXELN
Professional 350	P/OS

DECnet LAN's are based upon the Ethernet link protocol and support up to 1023 nodes in the DNA environment. Phase IV development efforts will enhance some of the DNA features including the support of up to 64,000 nodes in the network.

DECnet is an example of a propriety networking architecture supporting a homogeneous family of processors. Any combination of DEC computers and their associated native operating system is supported by DNA. Communication between DNA-based DEC computers and computers supplied by other vendors is accomplished by means of gateways. The DECnet/SNA Gateway enables DECnet nodes to communicate with IBM System Network Architecture (SNA) environment processors. The second alternative offered by DEC is the DECnet Router/X.25 Gateway. This enables DECnet nodes to communicate with processors which support the X.25 protocol on a packet switched data network.

DNA's architecture parallels the seven layer ISO model.

Although it is not specifically named an NOS by DEC, DNA does support many of the major high-level network functions characteristic of an NOS. Some of these functions include:

- o Task-to-task Communication,
- o Remote file and record access,
- o Terminal-to-terminal communication,
- o Network virtual terminal,
- o Problem isolation and network management,
- o Downline loading,
- o Upline dumping.

The salient implementation characteristics of some of these high-level functions are described below.

DECnet enables two programs or tasks running on different nodes in the network to exchange data over a logical link. Interprocess communication is implemented much like I/O operations; the logical link is similar to an I/O channel over which both programs can transmit and receive data.

DECnet's remote file and record access enables nodes to open/close files, create/delete files and read/write remote file records. Nodes may be different (DEC) processors running under different (DEC) operating systems. DNA/ISO layer 7 software enables users to invoke an interactive Network File Transfer (NFT) utility. A user may invoke NFT from any terminal and thus access and manipulate remote files. Network File Access Routines

Architecture Layers

ISO		DNA
Application	7	User
		Network Management
Presentation	6	Network Application
Session	5	Session Control
Transport	4	End-to-End Communication
Network	3	Routing
Data Link	2	Data Link
Physical Link	1	Physical Link

FIGURE 5.5: ISO-DNA ARCHITECTURE

Source: "Digital's Networks: An Architecture with a Future" - DEC

ORIGINAL PAGE IS
OF POOR QUALITY

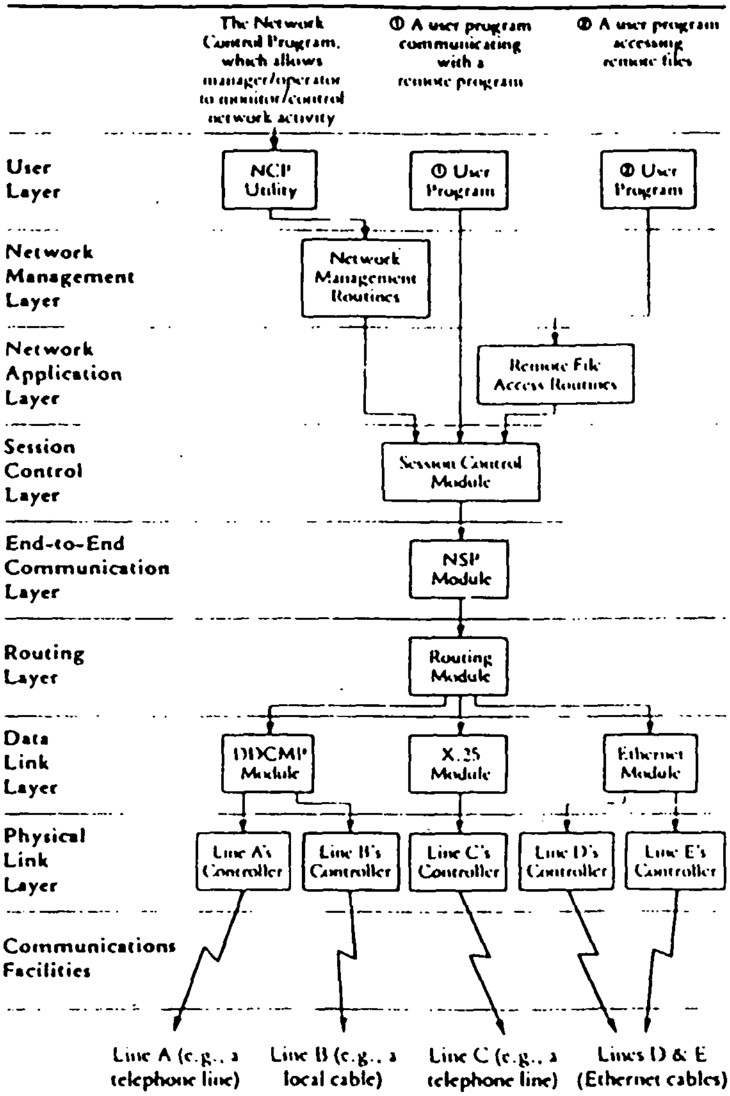


FIGURE 5.6: TYPICAL DNA MODULES

rce: "Digital's Networks: An Architecture With a Future" - DEC

ORIGINAL PAGE IS OF POOR QUALITY

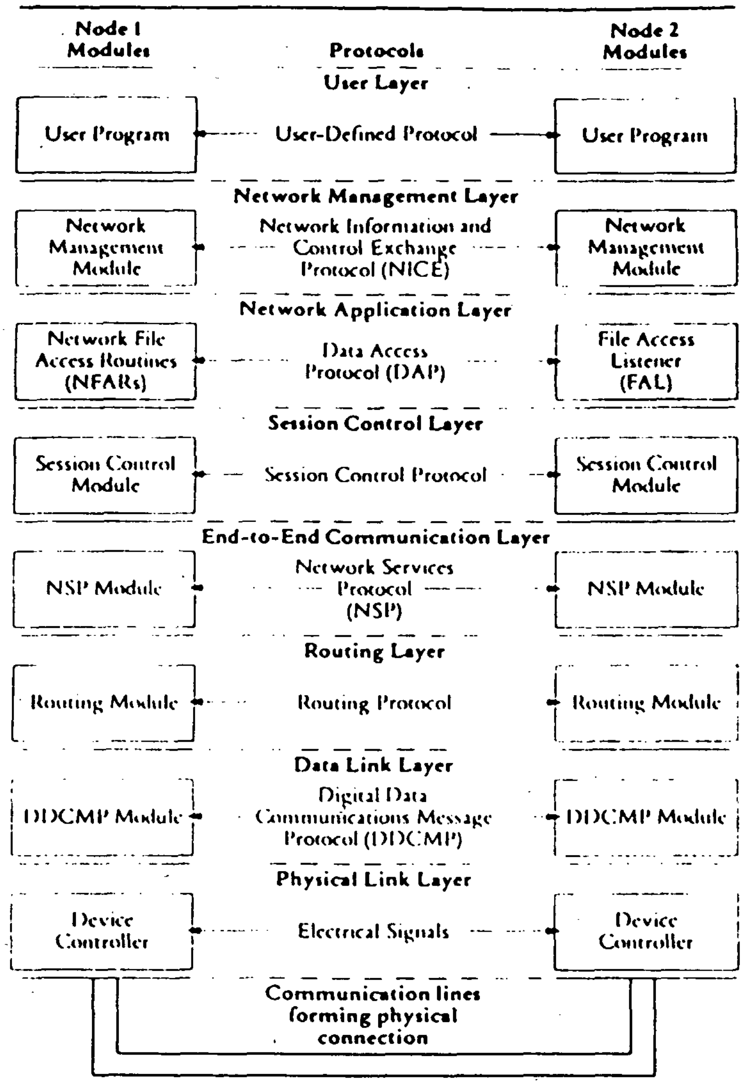


FIGURE 5.7: DECNET NODE-TO-NODE PROTOCOL COMMUNICATIONS

Source: "Digital's Networks: An Architecture With a Future" - DEC

(NFAR's) provide the DECnet user with a set of FORTRAN-callable subroutines that become a part of the user's process and enables access to remote files for application specific processing.

DNA supports several network terminal facilities. Users at two DECnet terminals can communicate interactively through Phone or TLK utilities. Phone is a utility of the VAX/VMS O/S while TLK is a DECnet utility supported by DECnet-RSX. One line/one way messages and dialogue nodes are supported.

The Network Virtual Terminal (NVT) facility logically connects a user at a local terminal to a remote node. DECnet Phase III requires that both the local and remote nodes run under the same O/S. Phase IV implementations lift this restriction. NVT services are handled by DNA/ISO layer. The Network Application/Presentation layers respectively.

DEC does support a UNIX operating system for the VAX-class computer. However, as of this writing, DECnet is not supported in the UNIX environment. During telephone conversations with DEC representatives, CTA was informed that this capability will very likely be supported by DEC by the end of 1985.

5.2.11 **QNX**

QNX 2.0 is a multi-user, multitasking real-time UNIX-based operating system supplied by Quantum Software Systems, Inc. of Ottawa, Ontario, Canada. QNX is designed for IBM PC and other processors which utilize the Intel 8088/8086 and 80186 chips. It provides a UNIX-like environment for single user configurations as well as configurations supporting a maximum of seventeen users. It does not depend upon any specific LAN technology.

QNX is a message based operating system. Tasks communicate by sending messages up to 64KB long. Inter-task messaging may take place between tasks residing in the same processor or in different processors on the network. An example cited by Quantum Software is the use of a message by a task to print data on a network printer. The task transmits a message to the Device Administrator task which prints the message and then returns a completion status message to the originator. The Device Administrator task need not be on the same processor as the originator since, to the originating task, the operation occurs transparently.

The Device Administrator handles all serial I/O devices for QNX. It is one of several administrators in QNX which are responsible for task creation and start-up, file maintenance, network communications, asynchronous time-out exception processing, and other functions.

The QNX network model is viewed as a collection of tasks executing on a collection of processors which have access to a collection of resources. QNX does not impose restrictions upon resources or processors available to a task. The key to accessing these objects is the idea of a unique node ID. For example, the command

```
p my_file > $lpt
```

prints a file to a local workstation printer. If this was to be printed on the printer located on node 3, the command would be:

```
p my_file > [3]$lpt.
```

QNX also supports remote task creation/execution. The command string

[1] sort < [2]/dirl/filel > [3]$lpt

causes a sort program to be executed on node 1, accept input from a file on node 2, and produce line printer output on node 3. Remote log-in (i.e., remote virtual terminal) is also supported by QNX. All I/O is defaulted to the requestor's terminal.

By means of bidirectional virtual circuits, QNX also supports the capability of intertask communication between processors. A user uses the system call

vc_create (nid, tid, buffer-size)

to create a virtual task ID supplying the node and task ID's, and buffer size of the largest message expected. Subsequent intertask communication takes place utilizing the virtual circuit ID's.

The messaging philosophy utilized by QNX may also be applied to provide for task synchronization. Tasks may be attached to "ports" which can then be used for inter-task handshaking via semaphores. Asynchronous interrupts, (i.e., exceptions) are also provided by QNX. Of the 32 exceptions available in QNX, sixteen are defined by the system (e.g., timer expiration, break, task kill) and sixteen are available for user defined conditions. This supplements the messaging technique for inter-task communication. In the QNX networking environment, QNX extends the exception processing capability to enable exceptions to be set on virtual tasks. This causes the exception request to be sent over the network and set on the real task running on another processor.

C-2

5.2.12 Apollo Domain

Apollo Computer, Inc. of Chelmsford, MA markets a token ring-board LAN product, the Apollo Domain, for its family of MC680X0-based workstations. The Domain system software environment is illustrated in Figure 5-8. Apollo offers two operating system environments for its Domain product: AEGIS, a proprietary O/S by Apollo and AUX, Apolla's implementation of UNIX. Both utilize Apollo's proprietary O/S kernel which provides virtual memory management and access to the Domain's high-resolution, bit mapped workstation displays and token-ring LAN. Salient features of AEGIS are listed below:

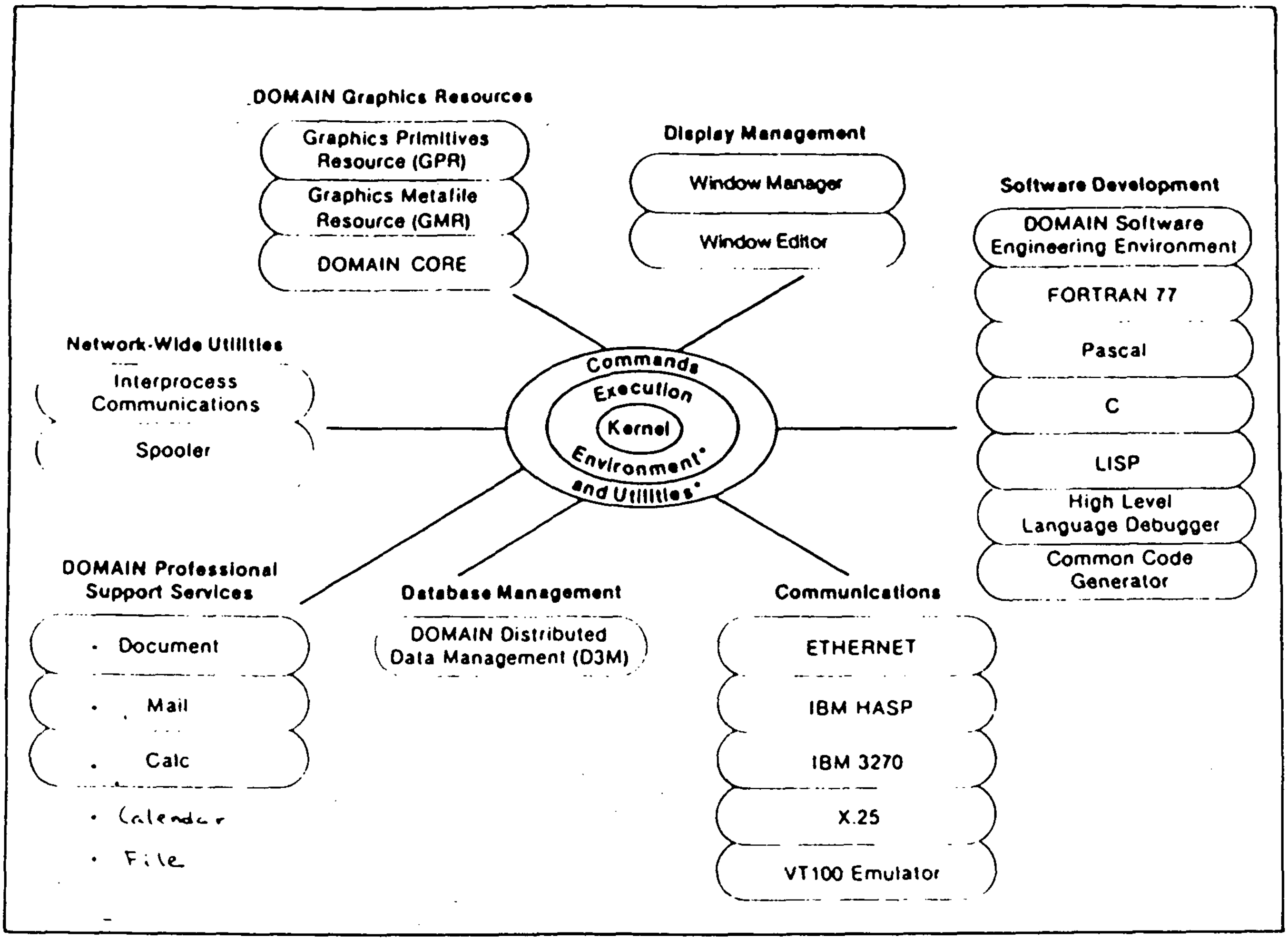
- o Virtual memory for direct execution of large programs
- o Network-distributed file system with access control list security and protection facility
- o Concurrent, multiwindow Display Manager Environment provides "virtual terminals" to programs, text, and graphics; includes screen-oriented editing
- o Interprocess communication, process creation, and event synchronization to coordinate execution of separate programs
- o Online HELP facility, including documentation of access to system services
- o Shell command line interpreter for application control: an interactive programming language whose "verbs" are user and utility programs
- o Logical, extensible set of system commands and utility programs
- o Support for a variety of programming languages and data management techniques
- o Graphics support
- o Support for diskless, shared server, and standalone equipment configurations
- o Network system reliability and maintenance utilities

The AUX O/S is based upon the Bell Labs System III Berkeley versions of UNIX. It supplements these features with:

- o support for I/O redirection, pipes, and forks,
- o support for Bourne and Berkeley C shell, and
- o the ability to initiate asynchronous processes

FIGURE 5-8 Apollo Domain Software Environment

ORIGINAL PAGE IS
OF POOR QUALITY



*Apollo Standard or Optional UNIX Command and Execution Environments

Both operating system environments offer demand paged virtual memory enabling users to execute programs which exceed the amount of physical memory in the user's computer. This capability has been extended across the Domain network.

The Domain is comprised of a homogeneous family of processors and operating systems which facilitates Apollo's support for programs to be directly executed on any node without recompilation or binding.

Apollo has integrated the network communications functions within the O/S kernel. A conscious decision has been made against a layered approach to achieve high network performance. Apollo has taken the opposite approach of LOCUS (see Section 5.2.1) in that only one copy of a file is maintained in the system. It may be shared by multiple users and Apollo justifies this design for the following reasons:

- it reduces the amount of mass storage devoted to redundant files and
- it eliminates the need to assure that copies of files reflect the same information.

Apollo has implemented an object oriented design in its approach for the Domain network. Each object is uniquely defined and made available across the network by a 96-bit virtual address (64 bits for a unique object name and 32 bits address within an object). Objects may be "typed" as text or graphics files, file security access control lists, database storage areas, and similar uses. They are byte addressable and can be located anywhere in the network.

Apollo's distributed network file structure utilized a hierarchical tree structure. A uniform naming convention permits access to any node, application program, data file, or resource in the network.

Access Control Lists (ACL's) have been implemented by Apollo for security access control to all objects. Objects may be accessed only when so established by an entry in the ACL. Combinations of user name, project membership, or node ID may be specified for authorized access control. Since the ACL also protects file system directories (an object), knowledge of a file's existence may also be denied if desired.

Apollo has implemented a server approach to promote resource sharing. Key to this is its implementation of interprocess communications (IPC). IPC permits users to communicate with servers via virtual circuits and it facilitates the exchange of messages between processes running in different workstations.

Outside the realm of the Domain network, Apollo also provides support for file transfers, remote virtual terminal, and virtual circuit services for X.25 communications. Ethernet in conjunction with TCP/IP supports file transfer and remote logon functions.

5.2.13 **XEROX**

XEROX Corporation has defined and published a suite of communications protocols, referred to as the XEROX Network System (XNS), that implement a Network Operating System. XEROX has incorporated these protocols in some of their workstation products, e.g., 8010 workstation. However, since XEROX has published these protocols, other vendors have implemented them in their products. These vendors include 3COM, Bridge Communications, and Associated Computer Communications.

The XEROX Network System and the DARPA protocol suite are similar in that both are:

- o independent of the local operating system,
- o based on connectionless internetwork protocol,
- o employ fewer protocol levels than OSI (the DARPA protocol architecture is not layered).

The architecture (shown in Figure 5-9) of the Xerox protocol suite is similar to the DARPA protocol suite which provides printing, filing, directory, and time of day services. The directory service known as CLEARINGHOUSE is distributed implementation used for locating both users and resources. For example, the user may select a file server and then the local workstation will invoke a CLEARINGHOUSE retrieve primitive to obtain the network address. Then, this address is passed to the transport layer to establish communication. It includes procedures for creating and modifying the directory as well.

The printing, filing, and directory service employ the XEROX Courier protocol, which is a remote procedure call. The X.410 standard is very similar to Courier indicating a significant XEROX contribution.

The transport and network layers are very similar to the DARPA protocol suite in functionality and operate over a wide variety of networks. However, XEROX also includes diagnostic protocols, ECHO in which the remote hosts return the bit stream entered by the user, and ERROR for reporting exception systems.

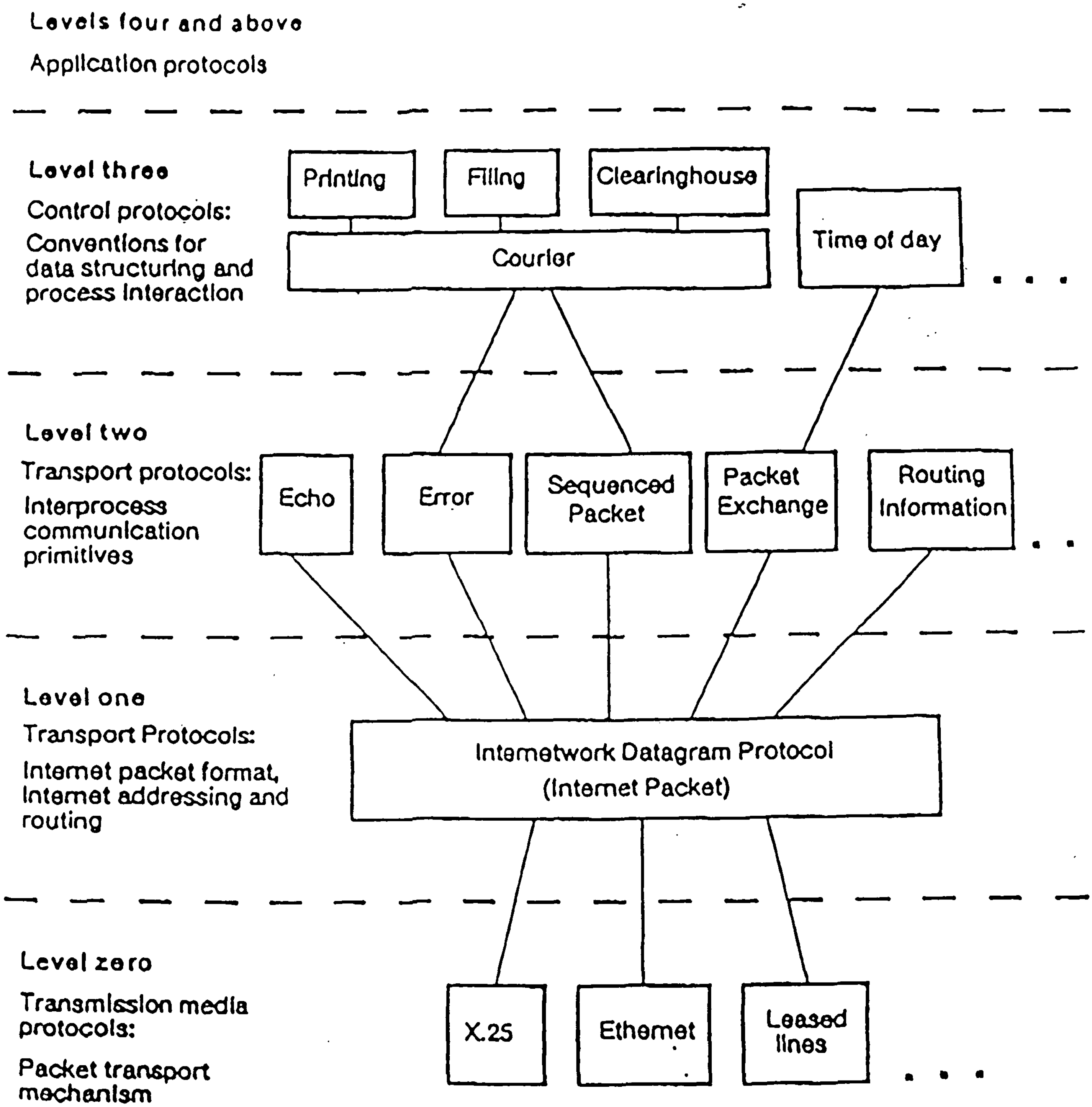

Levels four and above
Application protocols
Level three
Control protocols:
Conventions for data structuring and process interaction
Printing
Filing
Clearinghouse
Courier
Time of day
Level two
Transport protocols:
Interprocess communication primitives
Echo
Error
Sequenced Packet
Packet Exchange
Routing Information
Level one
Transport Protocols:
Internet packet format, Internet addressing and routing
Internetwork Datagram Protocol
(Internet Packet)
Level zero
Transmission media protocols:
Packet transport mechanism
X.25
Ethernet
Leased lines

FIGURE 5. 9

5.3 Research Projects

As illustrated in Figure 5-10, current research efforts can be categorized into the following areas:

- o real-time node performance including hardware and local operating system software enhancements,
- o job scheduling algorithms for optimal resource management,
- o reliability including fault tolerant systems and reconfiguration,
- o concurrency and recovery software procedures,
- o implementing new functionality in the NOS or implementing common functionality in new environments,
- o enhancement of programming languages to provide specialized features for concurrent programming and debugging of distributed software.

In the following sections, activities in each of the above areas are highlighted with references to specific activities particularly relevant to NASA.

A list of the research projects considered is included in the appendix. For each project, the following information is presented:

- name of project,
- organization,
- name of investigator(s),
- keywords identifying specific research areas.

The specific areas where the research efforts are contributing (beyond what is currently available in commercial products) are in concurrency synchronization, and recovery and

EFFICIENCY REAL TIME PERFORMANCE	JOB SCHEDULING	RELIABILITY	CONCURRENCY & SYNCHRONIZATION	FUNCTIONALITY	PROGRAM ENVIRONMENT
INTER-PROCESS COMMUNICATION	COST HEURISTICS	FAULT TOLERANT	SCHEDULING ALGORITHMS	REMOTE PROC CALL	INTERPROCESS COMM
	BIDDING	AUTOMATIC RECONFIGURATION		INTERPROCESS COMM	DEBUGGING
				NETWORK FILE SERVER	

FIGURE 5.10: RESEARCH AREAS

programming language enhancement, especially with the use of Ada (TM of Department of Defense).

To report some of the very recent results, CTA has relied upon session summaries of the January 1985 Workshop on Operating Systems in Computer Networks [46]. There were no proceedings of this workshop and hence, the technical discussion is necessarily brief.

The ARCHONS research project at Carnegie-Mellon is worth noting because of its scope and long-term duration. In particular, researchers are addressing many of the issues listed above, e.g., optimal resource management and high system availability for real-time control application (military combat platform, industrial automation). Their work addresses the problem from the establishment of a set of requirements (atypical for research) through implementation. The objective is to enhance system robustness and modularity which tend to dominate life cycle cost. The approach being taken is development of a tightly coupled system with a system wide operating system, which is "diametrically opposed to computer networks and network operating system". Hence, this significant project is not considered.

5.3.1 Real-Time Node Performance

The Real-Time Node Performance has been significantly enhanced with the continuing improvements in semi-conductor component performance. As the price/speed ratio of processors, memories, and other components continued to decrease, increased functionality can be incorporated into nodes and higher throughputs can be achieved. In fact, it is the rapid improvements in the semiconductor history which have triggered the advances in computer communications. These advances are now being illustrated in the commercial marketplace with implementation of specialized communications printed on circuit boards.

The primary activities in the local operating system support focus on more efficient interprocess communications. Work at IBM Switzerland [46] has addressed communication between OSI protocol layers such that data is being operated on by different protocol layers rather than being copied. These researchers claim that buffer management is facilitated because only application buffers are required and that no system buffers are required. Thus, with a single layer allocating buffers, flow control is simplified.

Work on the SWIFT project [46] at MIT has developed and implemented upward procedure calls for passing received data from lower layers to higher layers; these are referred to as "upcalls" by the researchers. The basis for this research is that procedure calls are more efficient than message passing between tasks or downward procedure calls that remain pending awaiting receipt of data. Some researchers argue that "upcalls" are simply software interrupts.

In the SWIFT project, TCP has been implemented using "upcalls". Some of the problems identified were harder to read code, security issues associated with passing data to less trusted higher layers, and garbage collection of the shared address space.

Work at Stanford on the Distributed V Kernel (not to be confused with the AT&T Unix System V) has focused on performance using interprocess communications. In this project, the researchers hide the intermachine communication from the user by providing an object-oriented interface to the users[50]. Since it is impractical to provide a universal set of objects to the users, they claim that the process group is the optimal set of objects to include in the interface because:

- processes are necessarily needed in the kernel,
- processes can support other types of objects such as files,
- processes are useful for supporting parallel competition,
- multiple processes easily support replicated processes.

In this work, they have incorporated message oriented process-to-process communication in the kernel and measured its performance. They have also built a file server employing this interprocess communication technique and intend to employ them for replicated data, job control, distributed naming, and multi-server transactions.

In subsequent work [51], they have investigated implementing a pipes feature, i.e., a one directional byte stream by either incorporating it into the kernel or building a server process using the interprocess communication service of the kernel. They have found that the latter approach was preferable because it was nearly as efficient and the kernel size could be minimized.

In its DISTRIX project, Convergent Technologies is developing a message based implementation of the UNIX System V for a network of workstations [46]. Note System V is the latest

AT&T supported version of UNIX. In this implementation, the user is unaware of the location of server processes, thus this is a distributed operating system mode of operation.

The implementation of this system is based on a real-time, message based operating system, the Foundation System. Presumably, this is proprietary to Convergent Technologies. The evolutionary development of DISTRIX has consisted of the following steps:

- implementation of UNIX as a single Foundation process.
- implementation of each DISTRIX process as a Foundation process.
- distributing each UNIX file as a DISTRIX file.

This project is significant because it would provide UNIX compatibility and alleviate performance problems associated with UNIX.

AT&T Bell Laboratories have been investigating the performance of UNIX based systems in a multi-processor environment. In this project, a VAX 11/750 running UNIX is off-loaded by Motorola 68006 processors running MEGLOS [52]. Users are connected directly to the VAX and the 68000's and VAX are interconnected by a high speed local area network.

By employing the UNIX TX command, a user can run a program on a specified 68000. The key features of this project are:

- extending the UNIX environment to the 68000 processors,
- efficient interprocessor communications based on full duplex channels employing message oriented primitives, however, these channels are pre-established and centrally

controlled,

- real processing executive in MEGLOS with subprocesses (which provides a multitasking capability) and semaphores for synchronization; the real-time capability provides efficient interfaces with disks, robot arms, video image processors, etc.

Performance measurements with MEGLOS indicate communications latencies of 750 microseconds and interprocess throughputs exceeds 300 kbytes, far exceeding standard UNIX capabilities.

5.3.2 Job Scheduling

Although there have been a number of theoretical formulations of optimal job scheduling, there is no algorithm available for optimal scheduling [22]. The problem can be addressed at two levels: entire job or task where multiple tasks comprise a job.

A typical approach for task scheduling is heuristic based on assumed processing costs for each task and interprocess communications costs. Using this information and possibly current system states data, the local system can execute an algorithm to optimally determine where the job should be executed. The problem formulations are usually very complex, employing team theory, control theory, or mathematical programming as discribed by Stankovic [22]. Because of the real-time requirements for execution, these algorithms must run very quickly; thus optimality is not usually achieved. These algorithms employ heuristics which tend to produce good, but not optimal solutions. Furthermore, significant overhead can be incurred in gathering status information. In practice, these heuristic algorithms typically have parameters which are left unspecified such that they can be tuned as system parameters for the local environment. This approach is used in the Apollo Domain structure [48].

Another approach typically used for job scheduling is the bidding scheme in which the local node issues a bid in which it queries other nodes regarding their capability and/or capacity to perform the current job. Based upon the responses received from the queried node, the originating node will assign the job to a particular node. These schemes are suboptimal and excessive overhead may be incurred in executing the bid process. There have been numerous research activities focusing on variations of this idea as summarized by Stankovic [48].

In the area of job migration, researchers at the University of California, Berkeley have developed a process migration (processes in execution) for the DEMOS/MP operating system. This operating system includes a message oriented interprocessor communication mechanism. The process migration implemented was dynamic in the sense that processes in execution could be moved. The criteria for deciding when a process should be moved was arbitrary, but adding a decision role to initiate such migration should not be difficult. The steps involved in performing this migration are:

- halt process execution,
- ask destination kernel for permission to move process,
- allocate process state in destination host,
- transfer the process state,

- transfer the program
- forward pending messages
- cleanup process state
- restart process.

The implementation of this capability was straightforward according to the researchers because the process state description was concise and not hidden in any functional modules in the operating system.

5.3.3 **Reliability**

This section addresses research in hardware fault tolerance and the area of concurrency, synchronization, and recovery.

5.3.3.1 Hardware

There is substantial work in the area of generic fault-tolerant computers which would be applicable to distributed computing systems. Work being performed by C.S. Draper Laboratories for NASA Johnson Space Center addresses this issue specifically for embedded computer systems.

In its Advanced Information Processing System, Draper is developing a fault-tolerant processor and a network node which is being implemented in a proof-of-concept network architecture. The node is employed as either an intercomputer network node or an I/O network node. The intercomputer network connects multiple fault-tolerant processors while the I/O network connects sensors, effectors, and general purpose computers to the fault-tolerant processors. In the proof-of-concept intercomputer network architecture, there are three independent subnetworks each having five nodes. Each of the five nodes in a subnetwork are connected to each of the other four nodes via a point-to-point port. The fifth port on the node is used to interface a host such as the fault-tolerant processor. Thus the intercomputer network provides a highly reliable means to interconnecting a set of hosts.

The significant characteristics of the Draper proof-of-concept intercomputer network are:

- o circuit switched connections between source and destination nodes;
- o point-to-point communication between nodes is employed, but the architecture could be enhanced to multiple access communication;
- o OSI protocol layers 1-3 are implemented in the OSI; there are no high level protocols implemented. Thus this system is essentially a LAN rather than a Network Operating System.

Key characteristics of the fault-tolerant processor are:

- o the architecture is based on two processors: a computational processor to perform protocol and applications processing and an I/O processor to perform I/O operations; the processors communicate via shared memory.
- o triply replicated hardware in the node with a voting algorithm to determine the "correct" result.

The intercomputer communications for the computational processor is simplified in AIPS because all remote devices appear to be memory mapped as shown in Figure 5.11. When a computational processor wants to send a message to a remote processor, it inserts the message into the appropriate location in the shared memory. Then the I/O processors (local and remote) and the intercomputer network deliver the message to the remote computational processor. The remote I/O processor inserts the

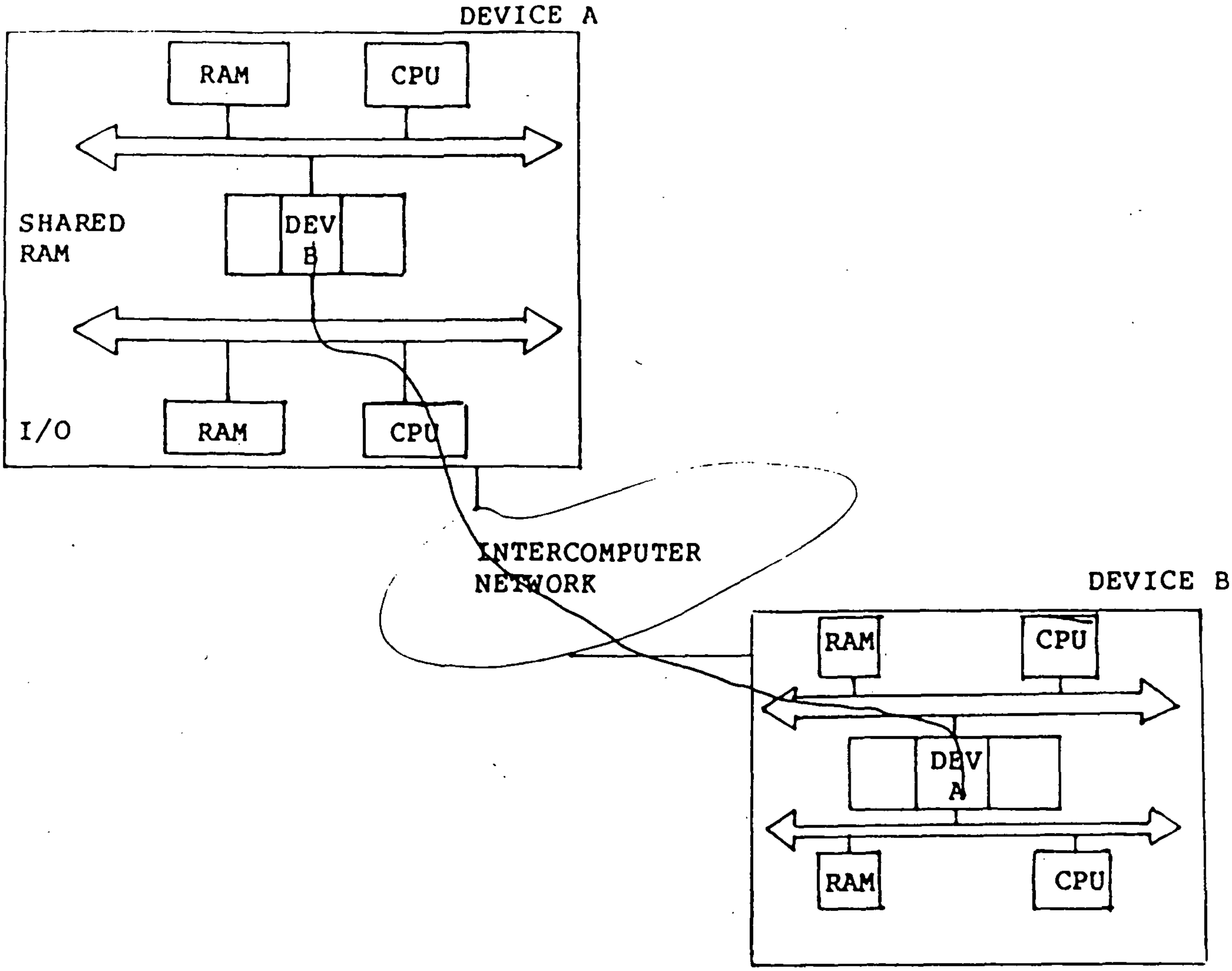


FIGURE 5.10: AIPS MEMORY MAPPED I/O

message in the appropriate location in the remote shared memory.

The major contribution of the Draper results is the fault tolerant hardware architecture memory mapped interprocessor communication rather than any specific networking or network operating system contribution. Apparently, Draper and JSC have future plans for incorporating high level protocol features into the AIPS, but details were not provided in the documentation.

Also, in the hardware area, the ENCHERE project [46] has developed a stable RAM with non-volatile memory banks to support failure atomicity and permanence for small objects. It has eight non-volatile memory banks mapped into the address space of the application processor. Hardware faults of a single bank are mapped by writing data into two banks (sequentially). The non-volatile RAM is also equipped with protection mechanisms to protect against illegal memory accesses.

The thrust of the CLOUDS project at Georgia Tech [5], partially funded by NASA, is to develop a reliable distributed operating system. The major contribution of this project is the development of the job scheduler with features to ensure reliable operation and to provide load leveling. The CLOUDS operating system has the following properties:

- multiple schedulers may reside in the same network;
- for each scheduler, there is a backup scheduler which is employed to detect failure conditions and assume control;
- servers, e.g., a print server, could be implemented across multiple hosts;
- processes communicate via remote procedure calls;
- use of the two phase protocol to ensure committment (see

Section 5.3.4).

The hardware components consist of multiple VAX and IBM PC hosts which are all connected by an ETHERNET. In addition, the VAX hosts are linked by a DEC CI interface. However, the initial distributed operating system work is focusing on the VAX. At this time, it appears that the IBM PC is used primarily for user interface.

5.3.4 Reconfiguration

Automatic reconfiguration has been implemented as part of the NASA Langley Research Center activities by employing multiple interfaces to multiple ETHERNET LAN's. In their network, a typical node had backplane connections to two ETHERNET communications units, one of which was connected to two parallel ETHERNETS while the other was connected to two independent ETHERNETS. Although multiple paths provide reliability, these researchers assign routes to messages based on delivery time requirements. For example, time critical messages are assigned shortest paths through the network.

In work partially supported by NASA, researchers at the SUNY at Stony Brook have developed the "UNCLE" algorithm for automatic topological reconfiguration of computer networks and theoretically analyzed its performance [69]. However, it has apparently not been implemented in their MICROS network.

In this work, the researchers consider failure of a node(s) in a tree shaped control hierarchy. When one of the non-leaf nodes (i.e., a node with at least one descendant) fails, then the control hierarchy must be reconfigured such that the orphans (descendants of the failed nodes) are re-connected to the control hierarchy. Thus, the reconfiguration algorithms must determine how to perform this reconnection subject to cost and performance considerations. The criteria employed in UNCLE are:

- localize the effect of the reconfiguration,
- distribute the reassignment of the disconnected nodes over more than one control node,
- do not increase the height of the tree.

In the algorithm, the father of the failed node acts as coordinator of the reconfiguration process and each orphaned node issues a message announcing the failure. The father node (grandfather of the orphans) performs a heuristic algorithm in which the reassignment of orphans is determined by the current computed loading. After the father determines the reassignment, messages are sent announcing the reassignment. A key feature of the algorithm is that it allows for detachment of nodes from the current hierarchy and reassignment to "UNCLES" in order to minimize tree height.

The theoretical analysis of the UNCLE algorithm has quantified total number of nodes with changes, change in tree length, and message traffic, which indicate the algorithm design criteria are achieved.

There has been substatial related work in the reconfiguration area for wide area networks including theoretical work with task force networks [62] as well as experimental work with packet radio networks [63]. Since the focus of this study is on local area networks, this area was not carefully surveyed.

5.3.4 Concurrency, Synchronization, and Recovery

The area of Concurrency, Synchronization, and Recovery is one where there is substantial research is being conducted. The focus of this research is to maintain data integrity when multiple users are attempting to access the data and device failures occur.

The OSI protocols address this issue and provide partial solutions. For example, some of the procedures included in the OSI protocols are:

- o retransmission at the link and transport protocol layers,
- o alternate routing at the network layer; NASA Langley has built an Ethernet network with the capability to provide this capability,
- o checkpointing and synchronization at the session layer, which are employed by higher layer such as the file service.

Also, there is the aforementioned work in progress to develop the Commitment, Concurrency, and Recovery standard at layer 7.

However, the OSI work is typically based on a connection between two parties. Thus, it needs to be enhanced to accommodate multiparty communication with processing and updates performed at multiple sites. The relevant research has originated in the data base community for handling file or database access. However, as Allchin and McKendry [45] point out, it is applicable in a more general programming environment such as updates of specialized queues, directory, storage allocation module. Although this was deemed an NOS application, this area was reviewed briefly for completeness.

The scope of the work in this area is comprehensively surveyed in a number of relatively recent articles:

Andres, G. and F. Schreider
"Concepts and Notations for Concurrent Programming",
ACM Computing Surveys
March 1983

Kohler, W.
"A Survey of Techniques for Synchronization and Recovery in Decentralized Computer Systems",
ACM Computing Surveys
June 1981

Bernstein, P. and N. Goodman
"Concurrency Control in Distributed Database Systems",
ACM Computing Surveys
June 1981

Also, there was a conference on this subject area which is documented in

Distributed Computing Systems Synchronization, Control, and Communication
ed. Paker, Y. and J. Verjus
Academic Press 1983

The fundamental concepts in this area are objects, actions, and atomicity. Objects are comprised of data and associated procedures acting upon them. An action (or transaction) is a partial order of operations on objects. The action is atomic if it can be viewed as either happening completely or not happening at all. However, the action is not implemented in an atomic way, i.e., computers could possibly fail during the execution of an action. Thus recovery and synchronization procedures are implemented such that the action appears atomic. The effect of an action on an object is not made permanent until the action is completed, this is referred to as committing. If a failure occurs or a user requests termination of an action during the execution of the action, then the action is aborted and the status of all objects reverts back to the original state as if the action was never initiated.

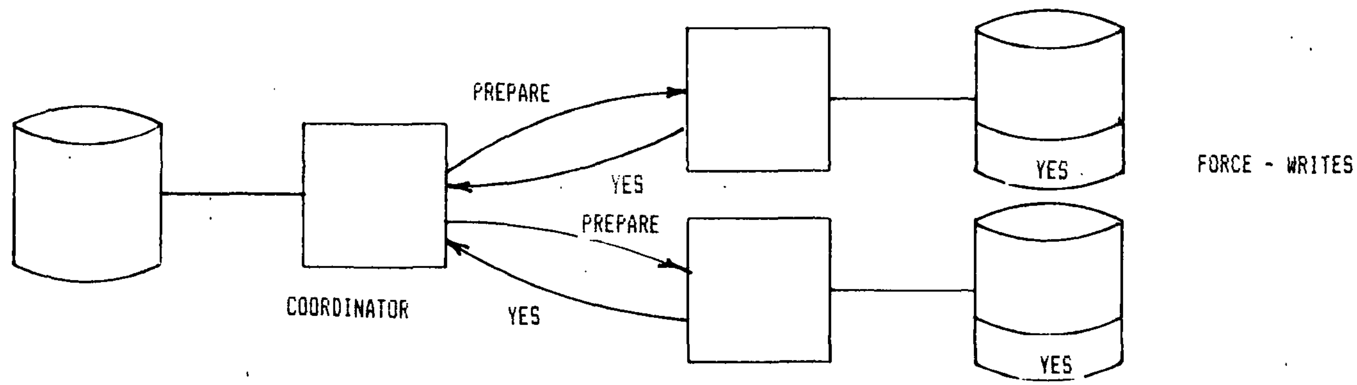
Another key concept is serializability meaning that a set of actions appear to execute in some sequential order even though they execute concurrently. Clearly, in order to provide acceptable response time to users, it would not be practical to execute all systemwide actions in a sequential manner. Bernstein [50] has done this fundamental work in specifying the conditions under which a set of actions are serializable. A serial set of transactions would introduce no concurrency or control problems because each action would be completed before the previous one finished. Since LAN actions are not going to be executed sequentially, the focus of the research is to develop action schedules that schedule transactions that are serializable. Thus, the correctness of a set of actions is judged by its serializability.

The basic protocol currently employed for effecting an update to a set of objects is the two phase protocol illustrated in Figure 5-12. In this protocol, all updates to the objects are tentatively performed and the centralized coordinator for the action initiates execution by transmitting a PREPARE message to all subordinates. If the subordinate is not prepared to complete execution of the action, then the subordinate returns a NO message. The coordinator will then abort the action.

If the subordinate is prepared to complete the action, then it writes a YES record to its log. Then it sends a YES message to the coordinator. If any NO messages are received, then the

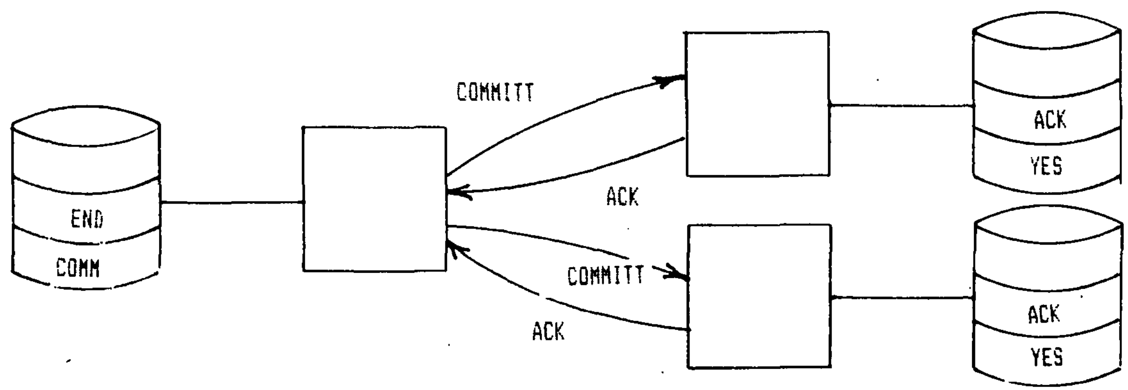
FIGURE 5.12: 2 PHASE PROTOCOL

1. TENTATIVELY EXECUTE TRANSACTION
2. SEND PREPARE MESSAGE



SUBORDINATE

3. IF ANY NO's, ABORT
4. IF ALL YES's, SEND COMMITT



5. ASSUME ANY FAILED DEVICE ULTIMATELY RECOVERS

SUBORDINATE

coordinator aborts the action.

If all YES messages are received, then the coordinator writes a commit record to the log and then sends COMMITT message to each subordinate. Upon receipt of the COMMITT message, the subordinate will write an ACK record and then forward a YES message to the coordinator. Upon receipt of all YES messages, the coordinator will write an END record to its log completing the action.

The handling of all failure conditions is tedious, but it can be established that the above protocol will successfully terminate under the assumption that all failed units ultimately recover.

Research topics in this area include:

- developing scheduling algorithms to ensure serializability,
- identifying conditions for scheduling algorithms which will provide correctness in a non-serializable environment,
- developing efficient techniques for implementing these techniques.

Recent work by Weihl [44] and Allchin and McKendry [45] have addressed the scheduling problems. Traditional scheduling algorithms have based their approach on the logical operations being performed to ensure concurrency. Weihl has extended this to develop data dependent scheduling algorithsm. In practical terms, Weihl's work is best illustrated by an example. Consider a bank account balance as an object with deposit and withdrawal operations. Traditional algorithms would not permit concurrent withdrawal operations because the first withdrawal operation could exhaust the funds in the account such that the second withdrawal could not be performed. Weihl develops the concept of dynamic atomicity which allow the concurrent withdrawal operating providing their sum does not exceed the current balance. Weihl carefully defines the concepts of dynamic and hybrid atomicity and proves a set of theorems to support his scheduling algorithms.

In the CLOUDS system, the researchers have implemented an object-oriented scheduler which has four levels of support facilities to enable users to achieve various levels of concurrency; these options are (in increasing order of concurrency):

1. Fallback to traditional locking schemes at the object level based on usual semantics of read/write.

2. Permit users to specify commutativity of object operations which must be true for all operations.

3. Provides programmer controlled locks which permits any item in the domain of interest to be locked.

4. User supplied procedures to specify to the scheduler object dependencies. The scheduler can then build a "template" object scheduler.

In their work, these researchers have developed theoretical conditions which enable use of these procedures.

Research work in this area addressing efficiency has been performed by Lindsay [47], who developed techniques to reduce the number of network messages and file accesses in special cases. For example, when a transaction is being performed across multiple nodes, some nodes may perform only read operations. Lindsay has developed modifications to the basic two phase algorithm to minimize the number of writes to the log and messages that must be delivered by the network.

5.3.5 **Functionality**

The functionality being implemented in research efforts, is very similar to that being implemented in commercial products, e.g., distributed file servers and remote procedure calls. In fact, many research projects are now offered as commercial products, LOCUS, The Newcastle Connection, Berkeley UNIX.

There are many research projects, implementing distributed file servers and remote procedure calls with specialized research objectives. Some typical projects include (and there are many others):

- ATHENA of MIT to interconnect many types of personal computers used for undergraduate education with IBM and DEC hosts,
- the IBM Carnegie-Mellon Information Technology Center project to provide a network-wide file system usable from personal computers providing interfaces and semantics of UNIX,
- Mayflower project developing connectionless remote procedure calls with binding done at call time by specifying a network address.
- NASA LRC project developing remote procedure calls with binding done at load time (although compile or runtime binding could be incorporated). This appears more like a process-to-process communication because the association is maintained rather than terminated after the exchange of data.

Some of the research projects have protocols for interprocess communication such as CHORUS (TM of INRIA) and MICROS. CHORUS [58, 64] introduces the concept of an actor, consisting of code and data. An actor can also send/receive messages but processing of such messages is sequential.

To send a message, an actor must open a port to establish an association with another actor. After the association is established, the actor invokes the send primitive service of the local operating system:

SEND (source port, destination port(s), message)

The operating system then takes responsibility for delivery of the message. Details of the message delivery protocol and exception condition handling were not specified. CHORUS is implemented on using CNET SM90 processors over an ETHERNET. The SM90 processors have been developed by the Centre National de Etudes des Telecommunications in France and employ Motorola 68000 or National Semiconductor 1600/32000 microprocessors.

The basic CHORUS system has also been enhanced with a number of fault-tolerant computing features. In CHORUS, a distributed application is implemented using actors distributed over the network. The key features of the architecture that provide the fault-tolerance are the introduction of protocls to provide:

- "coupling of actors" to provide backup computing resources over multiple sites,
- "activity message" to introduce a notion of atomicity and recovery beyond a transaction.

The activity message is used to define a protocol to synchronize and control the distributed processes.

MICROS [58] implements process-to-process communication using datagram transmission, i.e., all messages are transmitted independently. All possible sources and destinations/processes or devices, have global addresses that are pre-established.

Communication is implemented using mailboxes. The source process logically appends a header to the message containing source, destination, and length. The communication functions examine the mailbox to determine if the message is destined for a local or remote process. If it is local, then pointers are adjusted to pass the message to the local process. For remote processes, the message is transmitted over the communications network and then distributed to the destination process.

For high throughput applications, MICROS provides a service referred to as a "channel". Processes simply read from and write to pre-allocated buffers associated with the channel. The communications systems automatically sends messages corresponding to the pre-established buffer size.

The MICROS hardware currently consists of seven Heurikon HK68k Motorola 68000, two DEC LSI 11/23 systems, and a VAX 11/750. These systems are connected via an ETHERNET and the XEROX Networking System protocols at the higher layers. Although prototype networks are small, the goal of the MICROS project is to develop techniques applicable for very large networks, e.g., 10,000 hosts.

Also, at Carnegie Mellon University, the IBM LU6.2 has been implemented under UNIX to provide process-process communication. The interesting results of this project are [46]:

- o 85% of the protocol code was generated automatically,
- o protocol is large and cannot run in the Kernel,
- o simulating full duplex UNIX sockets using half duplex SNA connections; this required an extension of SNA.

In the area of data representation, Bolt, Beranek and Newman (BBN) has developed software to facilitate implementation of the CCITT X.409 Presentation Transfer Syntax [56]. This standard defines a technique for encoding exchange of information between heterogeneous computer systems. Example data types include Boolean, integer, bit string, and International Alphabet No. 5 string. In addition to specifying the encoding for the information to be transmitted over the communication network, this standard also specifies a notation for describing in a human readable representation the form and content of the information to be transmitted. The BBN software facilitates, both encoding and decoding of X.409 data streams.

They have developed a compiler, PRES, which converts modules of type definitions (specified according to the human readable form) ink type tables. The X.409 notation is input directly into PRES which:

- performs consistency checks on the data,
- generates type tables which are 'C' language source code files containing 'C' data declarations with initializers.

These tables are then compiled and made memory resident.

The remaining software consists of library routines to perform manipulations using file type tables. These routines include:

- o print_type, which prints a type table out in its original form as a type definition expressed in notation. This routine performs the inverse of Pres; it is used for checking the accuracy of Pres.
- o parse_value, which converts a value expressed in notation into a value table. Parse_value reads notation from a file, parses it according to information contained in a type table, and constructs a value table.
- o print_value, which prints a value table out in its original form as a value expressed in notation. This routine performs the inverse of parse_value.
- o encode_value, which converts a value table into a

sequence of octects encoded according to the X.409 encoding syntax.

- o a routine, decode_value, which converts a value, encoded according to the X.409 encoding syntax, into a value table.

Since the value tables are typically not in a computationally useful form, BBN has allowed the user to incorporate code fragments in the type spejcification indicating where the data is stored. Then the encoding/decoding routines can appropriately extract/enter the data from/to these locations.

5.3.6 Programming Environment

The major issue associated with programming languages is interprocess communication, especially with the use of Ada. The PULSE project [46] addressed this problem and considered the following alternatives:

1. Allow any Ada program to be arbitrarily distributed - this will only work if there is shared memory.
2. Distributed at the Ada task level and do all inter-machine communication via rendezvous.
3. Extend Ada with primitives for distributed computing.
4. Put separate Ada programs on each processor and build an inter-machine communication facility.

In essence, the issue is whether the Ada rendezvous technique used in local interprocess communication should be extended to remote interprocess communication. In this case, there would be a single Ada program with tasks in multiple nodes. This introduces substantial complexity and the Ada compiler must consider remote access. Alternatively, Ada could employ message-based communication based on OSI protocols for this interprocess communication. In this case, there would be separate Ada programs in each node. This issue highlights an issue that has been addressed for other programming languages in the past.

The PULSE project employed the standard approach and put separate Ada programs on each processor and built an inter-machine communication facility, which is the traditional approach.

Among the other distributed systems being implemented in Ada is the Cnet project at I.E.I. - C.N.R., Italy [46]. Although only limited information was obtained on this project, this effort will allow dynamic configuration to establish runtime linking of the new module. This requires a limited extension to Ada to allow for this runtime binding.

Experience has shown that debugging distributed software is extremely complex and has received relatively little attention. To facilitate distributed development, researchers for NASA Langley are developing specialized debugging techniques. The issues that are addressed are:

- o techniques for triggering event sequences; NASA work has implemented using an extension of the path expression in Path PASCAL [57].
- o monitoring and playing back the events occurring on the communications channel. This is typically done by employing a monitor node on the communications channel.
- o local debug/trace debug facilities in each node.

The MICROS project is also developing BUGNET, which provides similar capabilities BUGNET, which is written in MODULA-2, is designed to allow the programmer to control re-execution of software; it operates as a logic analyzer by saving all process interaction messages and checkpointing processes onto disk. To date, only a time-sliced version running under VMS on the VAX has been completed, but it is planned to develop a version for a network of MODULA-2 workstations.

At the University of California, Berkeley, researchers have enhanced Berkeley UNIX with a metering capability for interprocess communication [65]. As shown in Figure 5.13, processes A and B are communicating, but are also being metered. Associated with processes A and B, there are daemon processes which communicate with a filter process.

The measurement capability enables the user to determine when and how frequently calls are effected. The gathering of the information is performed by the Kernel and passed to the metering daemon process. The daemon meter process reports this information via interprocess communication to the filter. The filter will then record the information requested by the user. For example, the measurement capability could create a log of all messages sent between A and B for a user-selected set of message types. The recorded information would include source process ID, destination process ID, message type, time of event.

To facilitate debugging of distributed software, it is necessary to maintain global time such that relative timing of events can be determined. Berkeley UNIX researchers [2] have developed algorithms to estimate correct time based on observed delays in a single ETHERNET as part of their TEMPO time services. With clocks having movement of 10 ms, they determined that accuracy of global time could be achieved within an accuracy of 40 ms.

The technique that these researchers employ is to measure

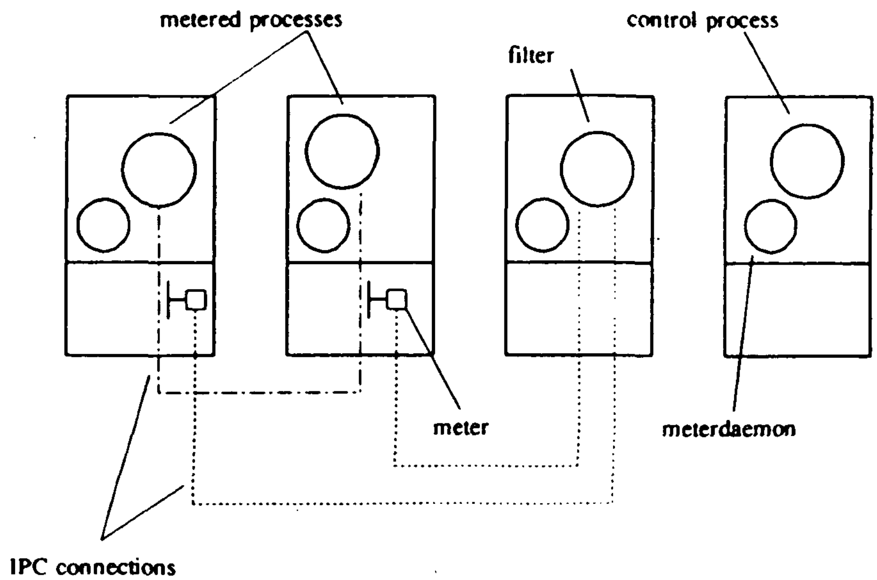


FIGURE 5.13: STRUCTURE OF 4.2BSD METERING TOOLS [65]

the delay from system A to B as

$$d_1 = TIMESTAMP_B - TIMESTAMP_A$$

where

$TIMESTAMP_A$ is a timestamp of the start transmission event entered at system A and incorporated in a control message sent to system B.

$TIMESTAMP_B$ is a timestamp at system B entered upon receipt of the control message.

The delay from B to A is analogously measured. Multiple experiments are made to determine observations

$$d_1(i),\ d_2(i),\ i = 1,2,...$$

Then minimum values are taken

$$d_1 = \text{minimum}_i d_1(i)$$

$$d_2 = \text{minimum}_i d_2(i)$$

The clock difference is estimated as

$$= d_2 - d_1$$

The use of the minimum rather than an arithmetic average was utilized in the above calculation in order to eliminate the effects of transmission delay variance.

The general algorithm to update clocks is as follows:

1. The master selects, using the Inter-Process Communication mechanism described above, one of the machines.

2. It calls measure and stores in an array the difference between its own clock and the clock of that machine,

3. After all the machines have been polled, it computes the network average delta as the average of all the different deltas.

4. It asks all the computers to correct their clocks by a quantity equal to the difference between the network average delta and their individual deltas.

Witte [66] has subsequently addressed the problem of correcting clocks and formulated the following alternatives:

(1) A single master can serve the entire network by polling all nodes for local time information, calculating the new times, and sending the new times to each node.

(2) Communicating masters can each keep time in a local area that does not overlap with others and can poll all other masters for input to the time calculation.

(3) Independent masters can keep time in overlapping areas so that each client node has fanout masters.

(4) All nodes can be autonomous with equal responsibility as master and client for its fanout neighbors.

Alternative (4) was selected as optimal because of its capability to accommodate nodes entering and leaving (including failures) the network.

These tools may facilitate development of distributed systems, but such development is still complex.

5.4 Research and Development Efforts

This section of CTA's survey addresses efforts of the research and development community relevant to this survey. The DARPA protocol suites has an historical significance and is described in Section 5.4.1. There are many commercial implementations of these protocols. Acceptance of the OSI reference model and use of OSI protocols in the General Motors Manufacturing Automation Protocol (MAP) is described in Section 5.4.2. The third effort, CRONUS, is a distributed operating system under development by Bolt Beranek and Newman, Inc. for the Rome Air Development Center (RADC), Griffis Air Force Base. This project is detailed in the last section (5.4.3).

5.4.1 Defense Advanced Research Projects Agency (DARPA)

The DARPA protocol suites are relevant to the state-of-the-art of Network Operating Systems because of their historical and operational significance. DARPA has support definition and implementation of a suite of protocols to implement a network operating system. There are numerous vendors that have implemented the DARPA TCP/IP protocol and offer it as part of a networking package; in particular, these protocols are included in Berkely 4.2 BSD. The DARPA protocol suite is a standard within DOD, but the National Academy of Sciences has recently recommended that DOD adopt ISO protocols at layers 3 and 4.

The similarities of the DARPA and XEROX protocol suite are that both are:

- o independent of the local operating system,
- o based on connectionless internetwork protocol,
- o employ fewer protocol levels than OSI (the DARPA protocol architecture is not layered)

The DARPA protocol suite illustrated in Figure 5-14 provides the virtual terminal, file transfer, and electronic mail servers as well as host servers providing time of day and internet addresses corresponding to a user supplied name. These services employ either a reliable transport protocol, Transmission Control Protocol (TCP), or user datagram protocol. The network protocol is the Internetwork Protocol (IP) which is a connectionless network protocol performing such functions as packet fragmentation and reassembly, packet lifetime control, as well as notification of security level, and network address. This protocol suite has been over a wide range of networks such as

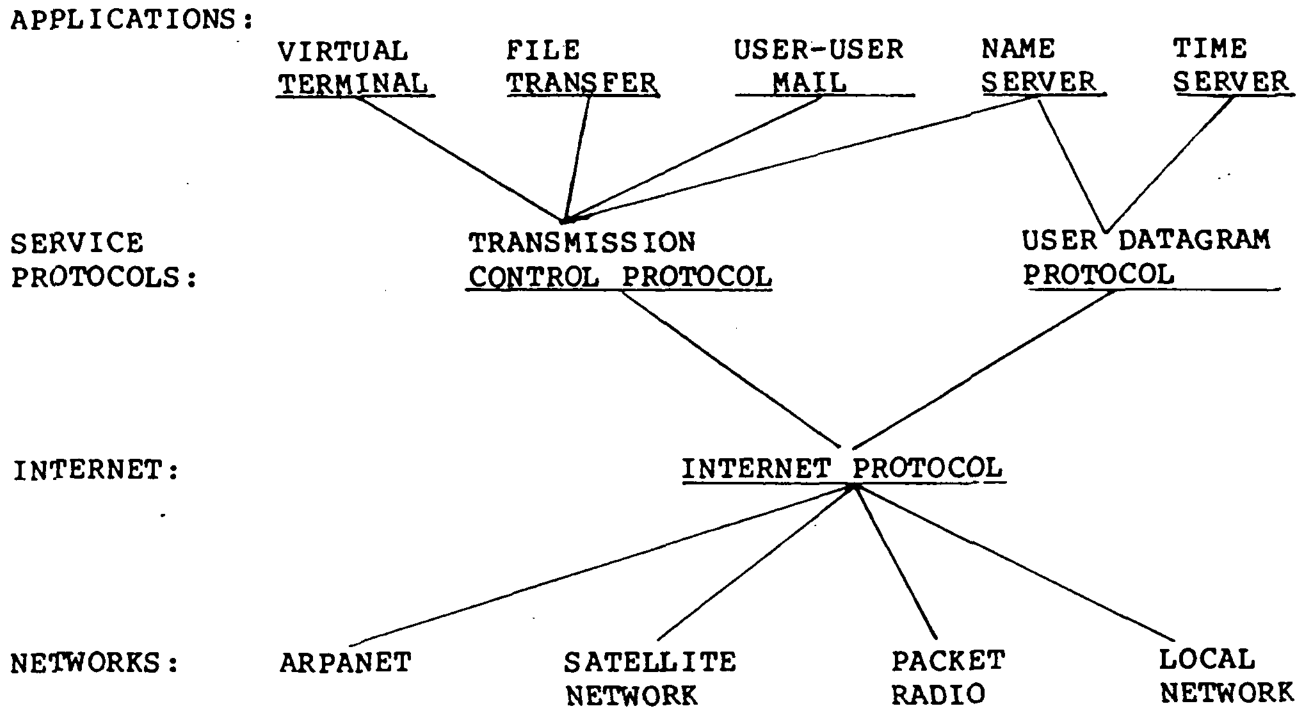


FIGURE 5.14: DARPA PROTOCOL SUITE

ARPANET, packet radio, satellite networks, X.25 networks, and local area networks.

5.4.2 Manufacturing Automation Protocol (MAP)

A significant event in the acceptance of the OSI Reference Model is the promulgation of the Manufacturing Automation Protocol by General Motors. Although the specifics of this standard are still evolving, it is clear that it will be based on OSI Reference Model and employ OSI protocols. Thus, OSI will make a significant penetration into a very large portion of the industrial automation market.

At the 1984 NCC, a multivendor demonstration was performed for a limited subset of the MAP. For this demonstration, a file transfer based on the ISO protocol (used with the ISO Class 4 transport) was performed over IEEE 802 type networks (token bus, carrier sense). Network and host components were supplied by different vendors, thus the successful file transfer empirically validated the concept of Open System Interconnection.

The architecture of the real-time network is still evolving, e.g., types of nodes, backbone network, access networks, bridge that will be required, etc. The protocols identified for inclusion of Version 2.1 are enumerated in Figure 5-15, but again the details of these protocols are still to be defined.

Although there are still many details to be finalized regarding MAP, it appears that many vendors will support it. Some examples of this support include (Systems and Software, March 1985, p. 30):

> DEC providing a UNIBUS interface for the VAX/VMS environment to a MAP token interface module.

FIGURE 5-15

MAP PROTOCOLS

1	IEEE 8Ø2.4 TOKEN RING
2	IEEE 8Ø2.2 LOGICAL LINK
3	ISO CONNECTIONLESS NETWORK SERVICE
4	ISO TRANSPORT
5	ISO SESSION
6	PRESENTATION
7	ISO COMMUNICATIONS APPLICATION SERVICE ELEMENT
7	FILE TRANSFER/ACCESS METHOD (FTAM)
OTHER	MANUFACTURING MESSAGE-FORMAT STANDARD (MMFS)
OTHER	APPLICATION INTERFACE
OTHER	DIRECTORY INTERFACE
OTHER	NETWORK MANAGEMENT
OTHER	PBX INTERCONNECTION

Texas Instrument announcing its TIWAY21 will be based on MAP.

Allen Bradley introducing VISTANET family of products, supporting MAP based on a 5MBps broadband network.

OEM Software products for layers 3 to 7 for the VAX/VMS environment.

Concord Data Systems providing a Network Control Computer and token SCOPE analyzer.

In summary, although still in its early development phases, MAP is a significant element of OSI activities.

5.4.3 **CRONUS**

CRONUS is a distributed operating system being developed for the Rome Air Development Center (RADC), Griffis Air Force Base by Bolt Beranek and Newman, Inc. (BBN). It is intended to promote resource sharing among computers and manage the collection of shared resources within a heterogeneous processor environment. One of its primary goals is to support the development and use of distributed applications. RADC, through its Distributed System Technology Program, has recognized the potential that a distributed systems architecture has for meeting future command and control requirements. BBN has divided the project into four phases: functional description, system design, system implementation, and test and evaluation.

Key to the CRONUS design (FIGURE 5-16) is an object-oriented view of the system. All system activity can be thought of as operations on a collection of objects managed by the system and organized into object types. Object types may be files, processors, devices, user records, and directories of catalog entries. Operations invoked on objects depend upon its type.

```
----------------------------------
|                                |



|  Manager Design Conventions    |
|- - - - - - - - - - - - - - - - |
|  Access Control Conventions    |
|- - - - - - - - - - - - - - - - |
|   Object Manager Standards     |
|--------------------------------|
| Request & Reply standards (OP) |
|--------------------------------|
|   Message Encodement  (OP)     |
|--------------------------------|
|  Message Passing Protocols     |
|small [packet] & large [stream] |
|--------------------------------|
|         IP  Protocols          |
|--------------------------------|
|              LAN               |
----------------------------------
```

FIGURE 5.16: CRONUS LAYERED ARCHITECTURE

The CRONUS system kernel contains an object-based interprocess communication facility which invokes operations on objects. This facility is designed to be host transparent. A CRONUS library will provide for a standardized interface to invoke operations on objects. This includes conversions to and from a standard data exchange format which facilitates interprocessor communication in a heterogeneous computer environment. CRONUS will implement an initial set of four object managers which will be used as building blocks to support CRONUS applications software. They are:

- o Catalog Managers - which, collectively, implement a system-wide symbolic name space.
- o Authentication Managers - which handle authorization and access control
- o File Managers - to implement the distributed file system
- o Device Managers - to permit the correction of terminals, printers, and other device resources.

Monitoring and control, and a user interface will be the two initial applications software implementations.

To verify the CRONUS distributed operating system concept, BBN will develop and implement a hardware testbed facility. The initial hardware components to be utilized in this testbed are a C/7Ø processor running UNIX, a DEC VAX running VMS, and special purpose microcomputers called Generic Computing Elements (GCE's). The GCE's are dedicated-function computers (all utilizing the same architecture) and support: file and data storage, user authentication, catalog management, device control and terminal access. The C/7Ø and DEC/VAX will be utilized as general purpose applications hosts and demonstrate processor and operating system heterogeneity. In addition, to test the idea of a dedicated processor for use as a C2 workstation, BBN will also augment the computer in the network with a Sun Microsystems workstation. Inexpensive personal computers (e.g., Apple, IBM PC) are under consideration for adding these processors to the network in the future. All computing elements will be clustered and networked utilizing an Ethernet LAN. Clusters will communicate using internet gateways. TCP/IP protocol will be implemented.

Host processor operating systems have been modified only minimally. Instead, the software necessary to integrate them into CRONUS runs as an adjunct to the host O/S.

6.0 Evaluation Relative to NASA Environment

The detailed NOS requirements for the NASA environment are obviously mission dependent, and during the course of this study such requirements were being formulated for Space Station as part of the NASA sponsored Data System Architecture Study (performed by teams led by the TRW and McDonnell Douglas.) However, high level NOS requirments can be formulated given the known operation in the space environment. The unique characteristics of the space environment are:

- diverse applications
- applications with stringent delay requirements
- applications with very high throughput requirements
- high system availability requirements

Nearly all of the characteristics discussed in Section 4 would be useful in Space Station. The notable exception is that terminal handling of classic TTY terminals probably would not be required. Instead personal computers employed as workstations would provide the terminal function (plus many other services) These workstations would access the host via a local area network rather being attached directly to the host. Because of their intelligence, the workstations would function more like hosts than classic TTY's.

Thus the CTA approach to addressing requirements is to evaluate how vendors provide the characteristics described in Section 4; this evaluation is discussed in Section 6.1. Then, based on this analysis, CTA identified deficiencies and associated technical development issues; they are discussed in Section 6.2. Finally having defined the broad range of requirements and assessed the state-of-the-art of NOS development, CTA prioritized implementation of the NOS features; this prioritization is presented in Section 6.3.

6.1 Evaluation of Features

CTA did not identify any NOS product which featured a comprehensive set of all of the NOS functional characteristics. Many products featured either:

- o a comprehensive implementation of a subset of the functional characteristics, or
- o a somewhat less comprehensive implementation of a larger subset of the functional characteristics.

User-to-user communications service requires that the NOS support inter-user communication and user-to-system communication. Services facilitating these capabilities include message handling, printing, terminal handling, and conferencing. CTA has indentified several products which support the first three services. The latter service (i.e., conferencing), however, was lacking in virtually all products. The electronic mail capability of the message handling service emphasized text and arbitrary binary data transmission. Graphics data transmission was not explicitly featured in the vendor products, however, bit-mapped displays in some cases would be handled. This is a technology dependent issue too since there are various mechanisms by which graphics data can be represented and subsequently displayed. Raster scan, storage tube, and refreshed vector monitors are the three principal technologies in use today and require different methods with which to represent the displayed data. None of the products offered supported the capability for facsimile or voice data transmission. Packetized voice and digital facsimile are at the forefront of technology and thus these two capabilities are usually handled as independent transmissions. Print service dominated the message handling capability of virtually all NOS products identified during the survey. These were often handled by print servers which spool requests from users on the network. Some vendors permitted any workstation to act as a print server and this could be dynamically assigned by the user. On many of the PC based systems, the print server was a specialized processor dedicated to just that one function. Furthermore, these systems often limited the quantity of print servers to just one or two such systems. Most often their use was advocated for office environments which shared an expensive letter quality printer for final document preparation. Virtually all vendors offered terminal handling services, often with a windowing capability. Again this was primarily for presentation of textual data. Facsimile, voice, and graphics were not handled. Some of the more sophisticated systems offered a virtual terminal capability thus enabling a user to utilize his/her host workstation as a dumb terminal to access and utilize a remote host and its resources.

Job Migration was most often manifested by implementions for remote procedure call and interprocess communications services. Job transfer and manipulation services were conspicuous by their absence. Interprocess communications was evident in the extensions to the UNIX pipe mechanism offered by several vendors as well as a proprietary implementation by DEC in its DNA architecture. Remote procedure call (RPC) is most prominent in UNIX extensions. The extensions have been to the UNIX fork and exec calls. A description of the most comprehensive implementation may be found in Section 5 of this report for the Locus Computing Corporation product LOCUS. Yet another implementation has been done by Sun Microsystems' NFS product. Charles River Data Systems' UNOS/Universenet product line have also extended the basic UNIX feature to implement an RPC (see Section 5).

Data migration, and specifically File Service was a prominent feature of virtually all NOS products surveyed. Implementations were as basic as disk servers, which offered little if any protection of multi-user access conflicts, to file servers offering some security and multi-user protection features. File servers were found also to be at two ends of the spectrum: some vendors offered a single, dedicated file server whereas others offered a more sophisticated approach to enable users to access a file anywhere on the network. The LOCUS product is an example of the latter capability. In addition, many sophisticated systems (e.g., LOCUS) offer complete transparency such that the user does not know (or care) where the file is located. Locus, for example, also offers multiple copies of files on the system; automatically tracks and updates them; and assures the users access to the most recent version. UNIX based systems, (e.g., 4.2 BSD) have also improved upon traditional file structuring to offer improved performance in a distributed file system environment. Data representation services were minimal for most vendors surveyed. This was due primarily to the fact that most NOS products were for a homogeneous computer environment (e.g., PC's and compatibles; or DECNET) and thus required few data translation requests. Vendors supporting heterogeneous computer environments (e.g., NRC Fusion) did supply utilities to convert, for example, UNIX file formats into MS-DOS (and vice versa) compatible formats.

Network control is comprised of three elements: recovery service, network management, and directory service. Locus was once again conspicuous because it went to great pains to assure that its replicated, distributed file system maintained its integrity in the event of system failure. It assures this with its atomic update feature which makes an updated file available to a user only after all modifications have been successfully performed to the data set. In the event of a failure, the user may access either a copy of the file stored elsewhere or safely access the previous version of the file. Locus assures that a file is not contaminated by "partial updates". This feature is of particular importance to many processing environments, notably space station. It demonstrates at least one element of fault tolerance required in such an environment. Network management capabilities ranged from simple statistics gathering packages (e.g., number of packets processed, network speed) to dynamic configuration of the system.

Also lacking was an adequate directory service for most systems. Most products made the assumption that the user know's where the other services (e.g., print, file servers) were located. Locus' file transparency made the file issue moot since the user was not concerned "where" the file was as long as it could be accessed.

Common features to other services identified are the transport and network security. Some vendors have planned (e.g., (NRC Fusion) to incorporate TCP/IP or class 4 ISO transports, but most vendors have relied upon LAN technology reliability and have not emphasized this capability. Network security was also minimal to best. Most offered multi-user lockout access to file servers but terminal and file access was usually limited to password protection. Encryption (e.g. DES) of data files and/or transmission was not an "advertised" capability. Other potential forms for user authorization access was minimal at best.

6.2 Deficiencies and Issues

In this section, CTA identifies the functional requirements of NOS that are not available in the current NOS technology (as evidenced by lack of commercial products or demonstrations) also the technical issues associated with providing these features are discussed. Three major areas are addressed:

- portability and performance,
- adherence to OSI protocols,
- completeness of functionality.

6.2.1 Portability and Performance

In its review of the state-of-the-art in Network Operating Systems development, CTA has determined that there are major tradeoffs between portability of the NOS software and the NOS performance, especially in the UNIX environment.To achieve the portability objective, the NOS software could be implemented as an application (relative to the local operating system) and employ a standard set of system calls to invoke the services of the local operating system. There are two difficulties with this approach, namely:

- there is no standard set of operating system calls applicable for a range of computers from micros through mini's to mainframes.
- communications software implemention requires real time I/O features that may not be provided to applications by the local operating system.

There are three standardization efforts that are currently ongoing to establish an operating system interface:

- IEEE P1ØØ3 Committee standardizing a UNIX based interface,
- IEEE 855 Committeestandardizing a microprocessor operating system interface,
- ISO TC 97 SC 5 WG 7 standardizing Operating system Command and Response Language.

Unfortunately none of these efforts are close to establishing an accepted standard, but the IEEE P1ØØ3 effort is of the most interest because of the pervasiveness of UNIX in the current market.

Currently there are a large number of variations of UNIX (including the most popular versions AT&t Systems V, Berkeley 4.2, and Microsoft XENIX) which the IEEE P1ØØ3 Committee is attempting to standardize. The scope and purpose of this project per its charter are "to define a standard operating system interface and environment to support application portability at the source level".

This effort entails three major components:

1. Definitions - Terminology and objects referred to in the document. In the case of objects, the structure, operations that modify these, and the affects of these operations need to be documented as well. (Sample Term: Pipe, Sample Object: File Descriptor)

2. Systems Interface and Subroutines (C-Language Binding) including:

   A. The range of interface & Subroutines in the/usr/group document.
   B. IOCTL/TermIO
   C. IFDEF Specifications
   D. Real Time (Contiguous files, synchronization, shared data, priority scheduling, etc.)
   E. Device interface, including Termcaps/Terml0
   F. Job Control, Windowing
   G. Network Interface (but not Protocol)
   H. Distributed Systems
   I. Device Drivers
   J. Error Handling & Recovery

3. User interface issues, including:

   A. Shell, Command Set, Syntax
   B. Portability - Media/Formats
   C. Error Handling & Recovery

In all of these areas, consideration will be given to defining the impact on security, international usage (language and character sets, etc.) and application needs such as transaction processing.

Unfortunately network protocols (along with graphics, data base management, record I/O) are outside the scope of the current P1003 project. However this provides the opportunity to use OSI protocol primitives as the network protocols. Ultimately this offers the opportunity to employ the ISO OSI protocols in conjunction with UNIX.

The relationship between the NOS and the local operating system is depicted in Figure 6-1. As shown in the figure the NOS intercepts all calls that could have remote significance; typical

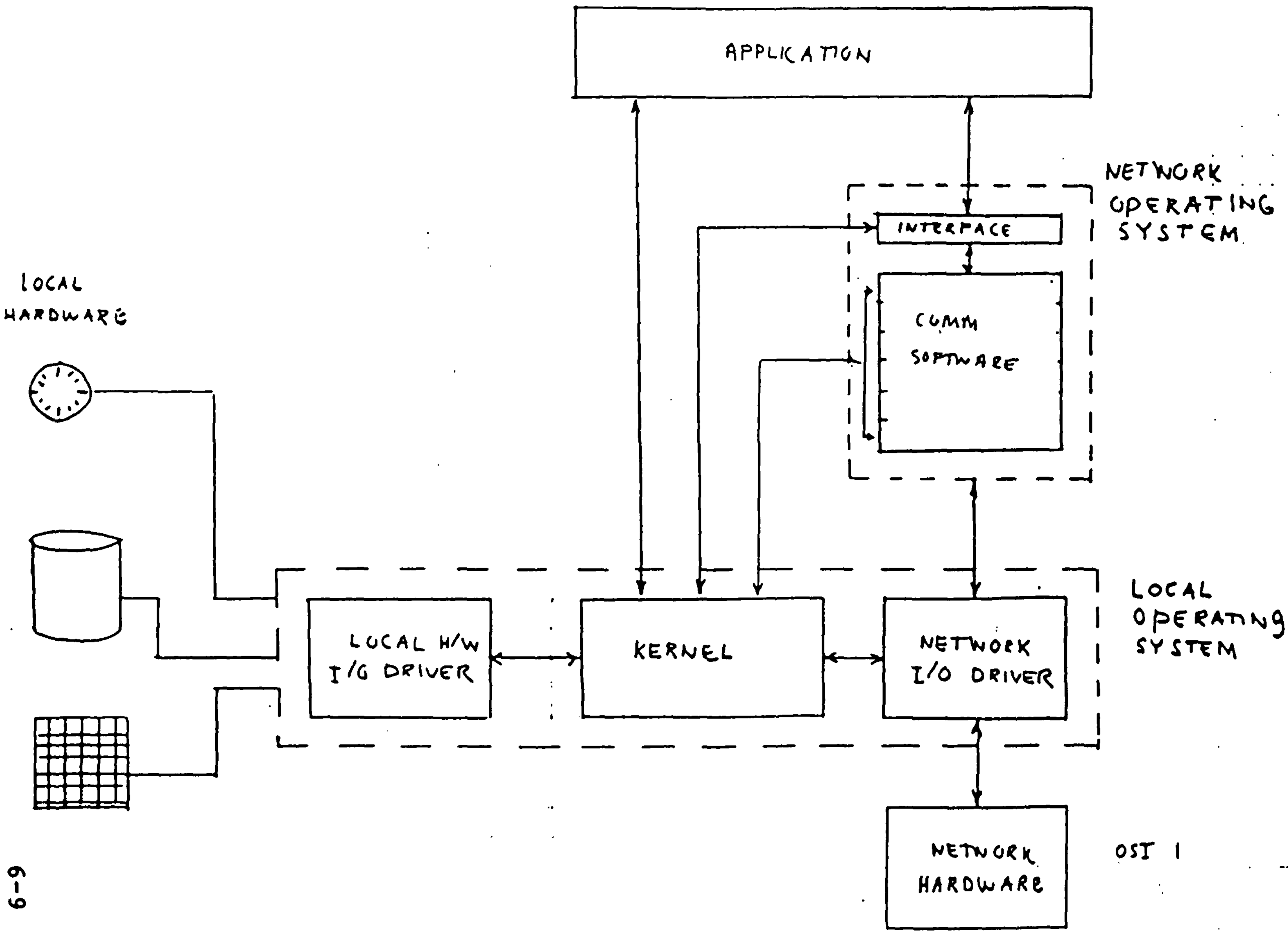


Figure 6-1 Software System Architecture

examples are file access and interprocess communication service requests. The redirector is responsible for processing these requests and determining if the request should be processed locally or remotely. There are some requests the the operating system that would always have local signiface, e.g. clock service requests to read the local time as to be interrupted often a specified time cust, memory management operations.

Note in the OSI environment the redirector must reside above layer 7 of the OSI and is only concerned with remote access. However in some cases such as the IBM PC, the redirector is shown to reside directly above layer 5, the session layer. In this implementation OSI layers 6 and 7 are null.

If the redirector NOS interface software determines that a service request can be handled locally, then the NOS interface passes the request directly to the local operating system. Then the local operating system with its local hardware I/O driver interfaces with the hardware.

If the redirector determines that a service request must be handled remotely, then the redirector passes the request to communications protocol software. As shown in Figure 6-2, this communications protocol software could be an implementation of ISO OSI protocol suite or an appropriate subset. This communications software would employ a network I/O driver to physically pass bits to the communication network. The OSI ISO layer 1 protocol is typically implemented in hardware as indicated in the figure.

Since the communications software employs a substantial amount of real time I/O, the keys to providing performance are

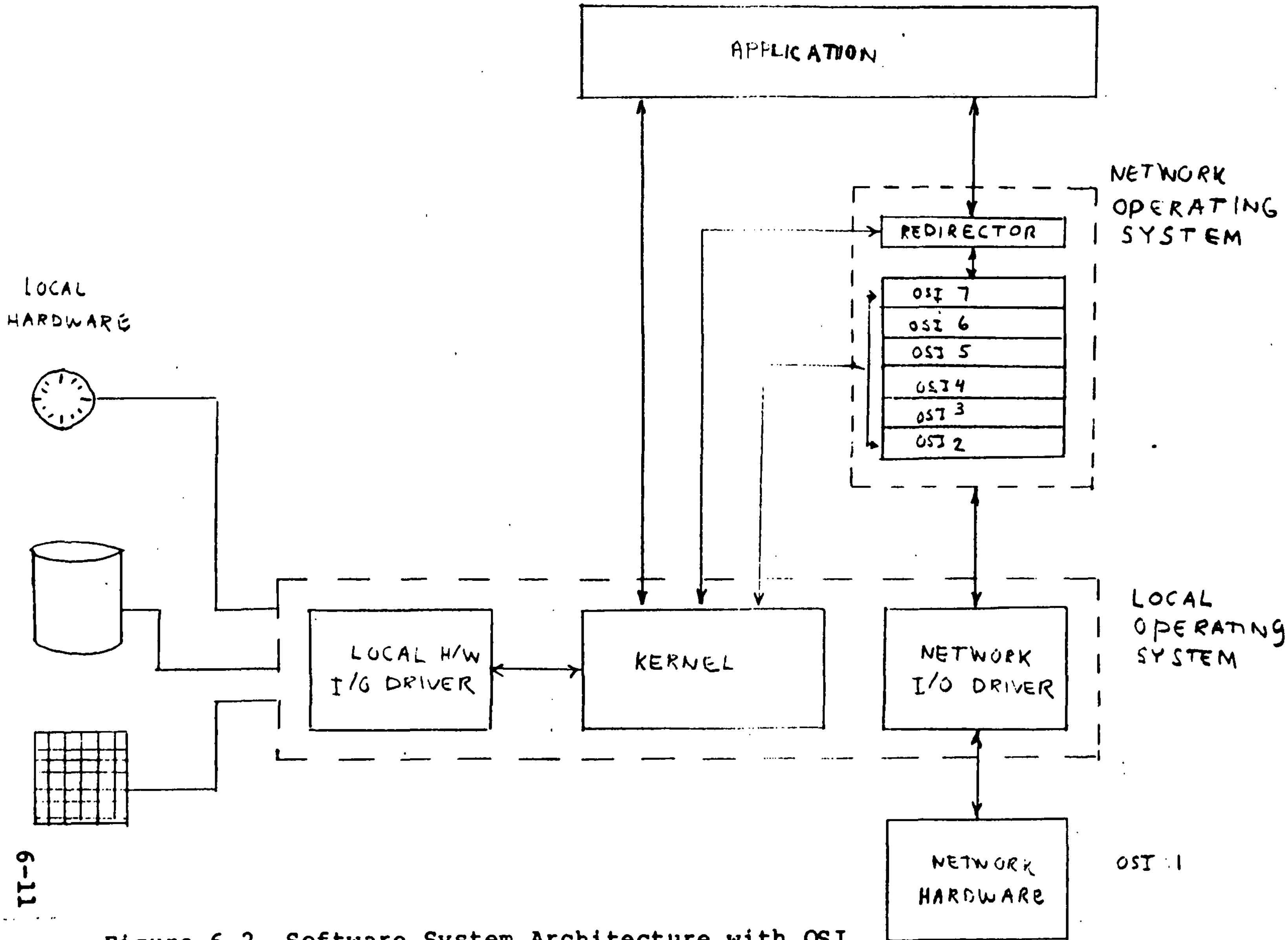


Figure 6-2 Software System Architecture with OSI

efficient means of interprocess communications and responding to interrupts. If adequate services are not provided by the local operating systems, then the NOS may have to be integrated with the local operating system to implement special services.

Typically the services provided by UNIX, which was originally developed for timesharing, do not provide adequate performance with respect to these requirements. Specific features that are required are:

- o message oriented interprocess communication (rather than byte oriented)
- o interrupt driven communication to alert a process that there is a message waiting for it (rather than implementing polled I/O)
- o semaphores to synchronize communication (rather than requiring a process to implement synchronization techniques)

More recent versions of UNIX (e.g. AT&T System V) do provide more efficient interprocess local communication using shared memory and semaphores.

The handling of interrupts associated with the real-time I/O may impose stringent scheduling requirements on the local operating system, which are typically not provided. Thus special servicing may have to be provided to the NOS.

Because of these stringent service requirements the local operating system and the NOS may be integrated. This integration may involve either:

- o special service request available only to the NOS or other local operating system modules,
- o access to specific local operating system data structures.

With these special requirements, the NOS is no longer an application, and the difficulty of porting the NOS is substantially increased.

CTA believes that the standardization efforts will not culminate in a useful interface within the needed time frame. These standardization activities should be tracked, but CTA recommends development of a standard operating systems with networking for NASA. This interface should be UNIX based complemented with a selected set of OSI protocol primitives. The actual implementation of the NOS with the OSI protocols is discussed in the following section.

6.2.2 Adherence to OSI Protocols

Although many systems and products have been developed, very few adhere to the ISO OSI Reference Model defining a seven layer protocol architecture. Even fewer vendors have implemented ISO protocols corresponding to the OSI layers. The latter point is not surprising since these protocols are just emerging as standards.

The lack of adherence to the basic OSI Reference Model can be explained in terms of the systems environment and requirements being addressed. Most of the development associated with network operating systems has been for local area networks where the developers have attempted to achieve performance transparency. In order to do so, they have developed simpler protocols with fewer layers. This is a reasonable approach because of the low error rate and low variability in transport delays. For example sophisticated transport protocols with capabilities to handle packets frequently received in error or occasionally arriving out of order, or having long transport delays are not needed. Similarly with only a single LAN, there is no need for any type of routing.

Another motivation for not implementing the OSI protocols is the memory requirements. To achieve high performance many of the UNIX networking capabilities are being integrated into the local operating system kernel. In fact most of systems developed are closer to distributed systems rather than network operating systems. To incorporate the ISO protocols into the kernel would make the local operating system memory requirements very large. Note IBM PC based systems build on the operating systems rather integrate the NOS into it.

For the problems currently being addressed, the approaches being implemented are probably optimal. However as problem complexity increases with internetworking of multiple LANs as well as interconnecting LAN's with Wide Area Networks, the ISO OSI protocols will become more appropriate and most likely necessary.

There are a large number of vendors developing OSI protocols. These products will likely be used in conjunction with local area networks. This will achieve the standardization and interoperability, but may still introduce a performance problem. To alleviate this problem, specialized communications printed circuit boards with faster processors will be developed.

Potentially, all seven layers could be off-loaded to a communications board with the redirector processing service requests to determine if they should be processed locally or passed to the communications board for remote processing.

However, there are some practical considerations. For example the file service needs access to the local file system, similarly the commitment, concurrency, and recovery service would need to access to the file system to store data that could not be stored in onboard memory.

Partitioning the NOS such that layer 7 resides on the local host processor and layer 6 resides on the outboard communications processor is feasible. However special attention must be given to the layer 6-7 interface because the presentation layer performs syntax conversions which are local operating systems dependent. It would not be desirable to make the communications processor local operating system dependent.

If layer 6 is implemented in the communications board and the layer 6-7 interface employs an Abstract Syntax Notation or X.409 encoding, then the communications board would be local operating system independent thus is a desirable implementation.

In the long term these are all potential solutions and vendors are beginning to provide products to implement them.

6.2.3 Completeness of Functionality

In this CTA presents the major functional deficiencies in current NOS technology; these include fault handling, remote interprocess communication, interoperability, conferencing, and directory.

6.2.3.1 Fault Handling

Response to fault conditions in the network is one of the serious deficiencies in Network Operating Systems development.

However, it is an area where substantial research is ongoing and recent OSI standards are incorporating features to handle these conditions. OSI protocol layers 1 to 4 will detect that a message cannot be successfully delivered, and it is the responsibility of the higher layers and the application to rectify the fault conditions.

As discussed in Section 4 one of the key issues associated with fault handling is the partition of responsibility between the NOS and the local operating system. This functionality is being incorporated in the OSI architecture at the:

- o session layer with the provision of checkpointing features, and
- o application layer with the committment concurrency, and and recovery protocol.

With the implementation of these protocols substantial progress will be made in the fault handling area. However this problem must be addressed in total including NOS and local operating system responsibility.

6.2.3.2 Remote Interprocess Communications

Interprocess communications service will be required to support distributed control programs executing in remotely located hosts. In this survey few vendors have been identified that provide this capability. The required features of this service are:

- o connection control to establish, maintain and release the connection
- o message oriented communication (as opposed to byte oriented communications provided by the original UNIX pipes)

- short delivery times (1 second as opposed to many seconds or minutes)
- very high throughput applications
- synchronization procedures to coordinate access to resources; this may be implemented via a semaphore or event counts.

In particular, the use of the Remote Procedure call to implement distributed control is not adequate because of deficient response times, especially in light of high throughput requirements. The remote program may not be loaded. Reading in the program from disk and loading it will take substantial time. Furthermore there is no ongoing dialogue between the local and and remote programs so program loading may have to be repeated with each communication.

Potential solutions for implementation are message communications enhancement to the UNIX pipes and the IBM LU 6.2 protocol.

Although the underlying hardware usually has a broadcast capability, multi-destination communication is not usually provided by the NOS. Instead, users must typically request transmission to each individual. This capability is lacking because of the complexity in handling the associated acknowledgement and retransmission protocols. However, a multi-destination capability has been incorporated in some research projects such as MICROS and STARMOD [67, 68].

6.2.3.3 Interoperability

As discussed above, developers have been categorized into two groups, UNIX and IBM PC based vendors. To date there has been very little work done to make these systems interoperable. The notable exceptions to this are the Intel OpenNet and Digital Research Fusion and PC DOS systems.

The most significant incompatibilities are the structure and the data representation services. In particular there are few efforts to distribute graphics or images between incompatible systems.

The purpose of Open Systems Interconnection is to provide this compatibility, but so far such interoperability has been achieved in practice only up to the transport level. With the development of presentation layer and file service protocols in particular, this interoperability will be achieved. Currenly it is technically feasible, but the requisite standards are still emerging and not yet widely accepted and implemented. Thus, although this is a current deficiency, it will be ameliorated.

6.2.3.4 Conferencing

A conferencing capability was not included in any vendor NOS product that was surveyed. However, conferencing capabilities have been developed in the research community and DECoffers a product VAXPHONE (TM of Digital Equipment Corporation) but it is not included in DECNET.

The lack of a conferencing capability is not viewed as a major limitation because voice communications can be used for conferencing. Ultimately voice communications may be integrated into the LAN further reducing the requirement for conferencing. However, there are still useful, but not critical, features of of the conferencing service (such as the conference record and templates for action items, resolutions, etc.) that could be implemented.

6.2.3.5 Directory

The lack of a Directory Service for locating users and resources in nearly all of the commercial products was surprising. Successful directory services have been implemented by XEROX (CLEARINGHOUSE) and DARPA (Name Server). In some cases such as the IBM Network feature, protocols are implemented to query the network upon demand to locate a resource.

The lack of the directory service for the prototype is a major deficiency.

6.3 Prioritization of Features

In this section CTA ranks the features of an NOS relative to their importance in the NASA environment. The prioritization of features is inherently a subjective judgement considering that the requirements set is not firm To prioritize features for implementation CTA has employed the following criteria:

- o importance of feature,
- o implementation complexity,
- o technical maturity.

Using these criteria has categorized the features of a Network Operating System into the following priority areas as shown in Table 6-1:

- o Connectivity features necessary to deliver bits through the computer network.

Table 6-1: Prioritization of Functionality

Connectivity

Layers 1 to 5 depending upon LAN functionality

1. **BASIC**

   File Service
   Message Handling
   Remote Procedure Call
   Print Service
   Directory Service
   Security
   Screen Interface

2. **Advanced**

   Level 1

   Process-Process
   Fault Isolation-Recovery
   Data Presentation

   Level 2

   Conference
   Statistics Gathering
   Job Execution Language

   Level 3

   Job Transfer and Manipulation
   Data Base Management
   Document Architecture

- o Basic features providing a broad range of essential capabilities implemented in a straightforward fashion which provide user-user, user-device, and process-process communication.
- o Advanced features introducing capabilites to make the NOS operation more sophisticated. These features, are further broken down by levels one (highest) to three (lowest) based on subjective evaluation of value added versus implementation complexity.

The connectivity features are absolutely essential in order to exchange information. However the scope of the functionality needed will depend upon the particular network over which the NOS will operate. For example if the network is a single LAN such as the Sperry Fiber Optic Bus Demonstration System or an Ethernet (TM of XEROX Corp.) then a reliable link protocol (between source and destination hosts) with a session protocol to provide the logical multiplexing of connections would be sufficient. However, if multiple local area networks are to be employed then protocol layers 3 and 4 would also have to be included. This may include a initial circuit protocol such as X.25 to preallocate resources in bridges, as well as ISO transport protocol. If the LAN is to be internetworked with a long haul network, then the class 4 transport protocol may be required with its packet retransmission schemes. In summary the specification of the protocols will be application dependent, but they will easily fit into the OSI architecture.

The Basic set of features would provide a wide range of services for user-server communication (file service and print service), user-user communication (message handling), and process-to-process communication (remote procedure call). For the file service and message handling service a complete set of the ISO protocols would not have to be implemented to achieve a very useful level of functionality. The set of specific features to be implemented is a subjective judgement, but a set of key features can be identified. Key features that would be required in the file service are:

- file sharing at the record level
- file ownership and access control-reading, writing, and modifying
- file locking to ensure concurrent updates are not made
- hierarchical file structure

The message service should include primitives for document submission delivery, delivery of copies, and probing of status. Also local in-box and out-box utilities should be provided to facilitate tracking documents transmitted or received.

The print service should include a spooling capability with elementary capabilities to recover printing from a specified point and routing of this print request to an alternative printer.

The remote procedure call is included in the Basic set because it would provide an elementary means of implementing distributed control. Furthermore it has been usually implemented and is offered with many systems. However, as discussed above it will not provide adequate performance for applications with stringent response time requirements. Thus a true process-process communication will be ultimately required.

Also the file service requires the definition of a virtual filestore. Since the on-board Space Station computer selection is a design issue, the virtual filestore and physical filestore could be identical. However, this would not address the interoperability problem with ground hosts or workstations which may not employ a compatible file structure.

Note that implementation of an arbitrary subject of features in a protocol does not conform to the guidelines for implementing the protocol. Thus it would not be possible to communicate with non-Space Station implementations, which implement a specific subject defined in the standard. This may be acceptable for a prototype implementation, but certainly not for deployment in an operational environment.

The Directory Service is included in the Basic features because of its utility to enable the users to employ the other services. For example users would be able to determine where available resources such as file services or applications are located or be connected directly (to them) as well as user locations.

Terminal handling is not a requirement in the NASA environment because PC's appear as hosts. However, a screen interface is required in order to display information on the user device. It would also include a user help feature such as a menu of commands and on-line user documentation.

Security features are essential to allow access to resources as permitted. It is envisioned that initially passwords would be implemented to enable users to access file or print servers and execute jobs on remote systems. These features are commonly available.

The advanced set of features included in level 1 are very important, but not quite sufficiently important to be included in the kernel of Basic features. The process to process communication function is very important, especially for distributed control applications. However, some rudimentary feature of distributed control could be implemented with the remote procedure call (although stringent response times could not be achieved) Thus the process-process communication is included in the advanced set. The specific process-process protocol is still to be determined but the IBM LU 6.2 is a possible candidate because of the lack of an ISO standard.

Fault Isolation and Recovery is also very important, but it is not included in the kernel because

- o Relatively low rate of failures in the LAN environment, especially with the Fiber Optic Bus Distribution System which has redundant network transmission media and interface unit components.
- o Research is currently on-going in the fault recovery area.
- o The uncertainty of acceptance of the ISO commitment, Concurrency, and Recovery standard.

However, as a first step in the NOS, checkpointing should be implemented as part of the session layer.

Data Representation is another very important feature but it is not included in the kernel because there will likely be a large degree of homogeneity in the initial implementation. This reduces the importance of providing data representation services. However, as heterogeneity is introduced, the data representation service will be required. It should be implemented with an incremental capability.

- o Static set of syntax tables for each user established in advance;

- o Negotiation of the syntax during establishment of the connection;
- o Dynamic switching of context during a connection; this would require implementation of the ISO presentation layer protocol.

In particular, communications of graphics will be required.

The features in level 2 of the advanced set are not nearly as important as the level 1 features, but they are useful additions. Conferencing is given a low priority because voice communications already provides a means of conferencing which will be both to both prototype and space station users. In fact, voice communications could possibly be accommodated by high speed LAN's such as the Fiber Optic Bus Demonstration System.

Statistics gathering was assigned a low priority because of its minimal utility in providing user services. However, it may be assigned a higher priority because of administrative procedures, e.g. accounting procedure may have to be implemented to effect chargeback to the users.

The Job Execution Language is a useful enhancement to enable users to employ the Remote Procedure Call. This would provide them with a means to specify:

- o on what processor a job should be executed,
- o when it should be executed, and
- o alternate procedures in case of fault conditions

Protocols in level 3 may be useful but are not recommended for immediate implementation. The Job Transfer and Manipulation Standard is complex and a useful capability can be provided with just the remote procedure call and a job execution language. It need not be implemented at all.

The Document Architecture and Distributed Data Base Management System (DBMS) are useful capabilities which build upon the Message Handling Service and File Service respectively. Thus they are defined as advanced features. Ultimately the DBMS service will be required and should be implemented while the requirements for Document Architecture are less clear.

APPENDIX A

-

Product Templates

(In Alphabetical Order by Vendor Name)

Advanced Computer Communications (ACC) - ACCES XNS

ORIGINAL PAGE IS
OF POOR QUALITY

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	ACCES XNS
2	Developer/Vendor:	Advanced Computer Communications (ACC)
3	Address:	720 Santa Barbara St Santa Barbara, CA 93101
4	Telephone No(s)/Contact:	(805) 963-9431
5	Documentation Availability/Source:	TBS
6	NOS Hardware Implementation:	
	---Network	Ethernet,others
	---Host Computer(s)	DEC PDP-11,VAX ; Charles River Data Systems processors
7	Host Operating System Name(s):	UNIX (System V, Version 7,4.2 BSD, UNOS) DEC RSX and VMS
8	Implementation Language:	TBS
9	Maintenance Support:	TBS
10	ISO Model Implementation:	Xerox XNS Architecture Network layer (L3) to Presentation (L6) Courier Protocols
11	List of Services for Each ISO Layer:	Network Management and Statistics Remote Procedure Call Data Representation Remote File Transfer Service
12	Major O/S Functions Implemented:	
13	O/S Modification(s):	TBS
14	Price:	TBS
15	User Added Applications S/W:	Applications programs written in C may interface with ACCES XNS protocol package

Comments:

1. Supports Xerox XNS communications protocol
2. File Transfer and Virtual Terminal support for XNS networks
3. XNS protocols supported:
   SPP - Sequenced Packet Protocol (transport layer of ISO)
   IDP - Internet Datagram Protocol (ISO network layer)

Alcyon Corporation - Distributed File System(DFS)

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	Distributed File System (DFS)
2	Developer/Vendor:	Alcyon Corporation
3	Address:	8716 Production Avenue San Diego, CA 92121
4	Telephone No(s)/Contact:	(619) 578-0860
5	Documentation Availability/Source:	Source code available
6	NOS Hardware Implementation:	
	---Network	Ethernet
	---Host Computer(s)	MC680X0-based workstation
7	Host Operating System Name(s):	REGULUS (compatible with UNIX V6,V7,SIII,SV)
8	Implementation Language:	"C" (for DFS)
9	Maintenance Support:	TBS
10	ISO Model Implementation:	not implemented
11	List of Services for Each ISO Layer:	n/a
12	Major O/S Functions Implemented:	Remote file access Remote program execution
13	O/S Modification(s):	REGULUS is a proprietary o/s
14	Price:	$2400 - single copy binary w/C compiler $22000 - single copy source w/C compiler Price of DFS is (TBS)
15	User Added Applications S/W:	TBS

Comments:

1. DFS protocol is based upon DELTA-T packet transport protocol developed at Lawrence Livermore National Laboratory. It is described in "DELTA-T Protocol Specification" by RW Watson 4/15/83.
2. Heterogeneous CPU environment is possible with DFS
3. File and Record locking supported
4. Diskless workstations possible since all file I/O may be done at remote workstation w/o the need to transfer file to local workstation
5. Future implementations planned for DFS:
   - DEC VMS support
   - Data General AOS support
   - 4.2BSD UNIX support
   - AT&T System III support
   - Lantech uNETix support

Aoollo Comouter. Inc - Aoollo DOMAIN

ITEM #	ITEM	DESCRIPTION
1	Name/Availabilitv of NOS:	Aoollo DOMAIN
2	Develooer/Vendor:	Aoollo Comouter. Inc
3	Address:	15 Elizabeth Drive Chelmsford. MA 01824
4	Teleohone No(s)/Contact:	(617) 256-6600 Donovan Buell. Jr.
5	Documentation Availabilitv/Source:	TBS
6	NOS Hardware Implementation:	
	---Network	Token Rino LAN
	---Host Computer(s)	Aoollo workstations - Motorola M680x0-based
7	Host Operatino Svstem Name(s):	AEGIS (proorietarv) AUX (UNIX - based)
8	Imolementation Lanouaoe:	"C"
9	Maintenance Suooort:	TBS
10	ISO Model Implementation:	Not imolemented - uses non-lavered proprietary approach
11	List of Services for Each ISO Laver:	N/A
12	Major O/S Functions Implemented:	Real-time,multitaskino Message handling,print service,terminal handling remote procedure call,interprocess comms file service network manaoement,network securitv
13	O/S Modification(s):	AEGIS is a prorietarv O/S; AUX - proprietary UNIX implementation
14	Price:	Servers: $9750 - $56,500 Computational Nodes: $9900 - $74000 AUX $300 per node; $6500 per site (up to 100 nodes) see vendors comprehensive price list for detailed pricino information
15	User Added Applications S/W:	Available via software develooment tools under AEGIS and AUX O/S's

Comments:

ORIGINAL PAGE IS
OF POOR QUALITY

AST Research - PCnet-II

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	PCnet-II
2	Developer/Vendor:	AST Research Inc.
3	Address:	2121 Alton Ave Irvine, CA 92714
4	Telephone No(s)/Contact:	(714) 863-1333 Debbie Hassett
5	Documentation Availability/Source:	Users Manual Installation Manual Print Spooling Manual Technical Reference Manual
6	NOS Hardware Implementation:	
	---Network	twisted pair,baseband bus (CSMA/CA)
	---Host Computer(s)	IBM PC,XT,AT, or portable
7	Host Operating System Name(s):	PC-DOS 2.0,2.1,3.0,3.1,Microsoft Networks 1.0
8	Implementation Language:	Assembly and "C"
9	Maintenance Support:	board return or exchange no field maintenance s/w updates provided
10	ISO Model Implementation:	Does not conform to ISO model
11	List of Services for Each ISO Layer:	File sharing, print spooling
12	Major O/S Functions Implemented:	File, record security protection
13	O/S Modification(s):	TBS
14	Price:	$495 and up
15	User Added Applications S/W:	TBS

Comments:

1. Comms boards by Orchid Technology
2. Proprietary CSMA/CA protocol
3. Uses reliable datagram service
4. SNA/BSC gateway w/3270 communication
5. PC's can access up to 16 disk volumes
6. Dedicated file and print servers are not required, Shared PC's can also function as workstations.

Bolt, Beranek, Newman - CRONUS

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	CRONUS
2	Developer/Vendor:	Bolt, Beranek, Newman (BBN)
3	Address:	10 Moulton Street Cambridge,Mass. 02238
4	Telephone No(s)/Contact:	Richard E. Schantz - principal investigator (617) 491-1850
5	Documentation Availability/Source:	BBN Report #5885 - CRONUS, A Distributed Operating System: Phase 1 Final Report January 1985 BBN Report #5879 - CRONUS, A Distributed Operating System Functional Definition and System Concept June 1982 (revised 1/85)
6	NOS Hardware Implementation:	
	---Network	currently Ethernet
	---Host Computer(s)	DEC VAX,C/70, Sun Microsystems workstation
7	Host Operating System Name(s):	UNIX(C/70), VMS(VAX), 4.2BSD(VAX)
8	Implementation Language:	"C"
9	Maintenance Support:	see note 1 below
10	ISO Model Implementation:	not ISO, but a layered approach
11	List of Services for Each ISO Layer:	n/a
12	Major O/S Functions Implemented:	Object management Standard interprocess communications facility System wide distributed file system System wide symbolic ame space for all types of objects Facilities for process management User and process authentication Standard access control discipline for all object types User interface for all CRONUS and applications services Monitoring and control services
13	O/S Modification(s):	TBS
14	Price:	not applicable - see note 1 below
15	User Added Applications S/W:	TBS

Comments:

1. CRONUS is a research project currently in progress at BBN Labs for the Rome Air Development Center Griffiss Air Force Base
2. Hetergeneous clusters of cpu's supported
3. Use of object oriented design
4. "The CRONUS distributed operating system is intended to promote resource sharing among interconnected computer systems and manage the collection of resources which are shared. Its major purpose is to provide a coherent and integrated system based on clusters of interconnected heterogeneous computers to suppport the development and use of distributed applications." BBN Report 5885

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	UNIVERSE-NET
2	Developer/Vendor:	Charles River Data Systems
3	Address:	983 Concord Street Framingham, Mass. 01701
4	Telephone No(s)/Contact:	(617) 626-1000 Dick Swee, Manager Network Software & Operating System Development
5	Documentation Availability/Source:	List TBS
6	NOS Hardware Implementation: ---Network ---Host Computer(s)	 802.3(Ethernet) & 802.4(Token Ring) PC,NCR,Honeywell,Intel,CRDS Universe (68000-based)
7	Host Operating System Name(s):	UNOS and UN/System V (UNIX-based)
8	Implementation Language:	"C"
9	Maintenance Support:	Built-in test facilities
10	ISO Model Implementation:	Implements all 7 ISO/OSI layer protocols
11	List of Services for Each ISO Layer:	L6 File system interface L7 Electronic mail,messaging(UNIX Write command), Printer sharing,virtual terminal
12	Major O/S Functions Implemented:	Interprocess communications,file sharing, electronic mail,messaging,printer sharing, virtual terminal,file transfer using draft ISO FTP
13	O/S Modification(s):	Network OSI layers 1-5 are included in the O/S by linking the network code with UNOS kernal to optimize performance and allow faster access to system functions. Network is linked to UNOS as a standard driver so that the network can be called as if it were using a driver using standard OPEN, CLOSE,READ, etc.
14	Price:	Approx. $1500 per node
15	User Added Applications S/W:	Supported as an initial objective of the product. Easy addition of different network media via addition of UNOS device driver.

Comments:

(1) Session control functions are more available. Thus applications will tend to use these functions rather than duplicate session functions in each applications program.

(2) Since network code is linked to UNOS and is treated as a device driver, network access is provided to programs and file system in a manner identical to traditional device access.

(3) Supports both local(LAN) and wide (WAN) area networks

(4) Built-in test facilities available.

(5) UNOS developed to support real-time and networking environments. UNOS supports:
...Synchronization,prioritization,& allocation of resources
...Resident locking of processes
...Preemptive priority scheduling
...User-controlled priorities
...Fast interprocess context switching & interrupt service
...Generalized exception processing
...Synchronization and scheduling mechanisms
...Designed for multi-user systems

Corvus Systems - Omnishare.Omninet

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	Omninet.Omnishare
2	Developer/Vendor:	Corvus Systems
3	Address:	2100 Corvus Drive San Jose. CA 95124
4	Telephone No(s)/Contact:	(408) 559-7000 Tom Dowd - sales Bruce Byrd - technical
5	Documentation Availability/Source:	"System Manager Guide" "Mass Storage General Technical Information" "Omninet GTI"
6	NOS Hardware Implementation:	
	---Network	RS-422 twisted pair CSMA/CA
	---Host Computer(s)	PC's or compatibles,Apples DEC Rainbow, TI Professional
7	Host Operating System Name(s):	PC-DOS 3.0
8	Implementation Language:	USCD - P system
9	Maintenance Support:	s/w updates for bug "fixes" only h/w board exchange,800 number for support, and on-site consulting available
10	ISO Model Implementation:	Unknown
11	List of Services for Each ISO Layer:	TBS
12	Major O/S Functions Implemented:	Printer sharing,log-on security
13	O/S Modification(s):	no mods to local o/s's system volumes are added to each drive
14	Price:	CSMA/CA transporter card about $500 per node software about $150 disk drives (5MB - 126MB) about $1500 - $5000 print server about $900
15	User Added Applications S/W:	TBS

Comments:

1. Share printers through a dedicated print server.
2. Some file security
3. Read only and read/write access to public volumes
4. User ID and password log-on
5. SNA gateway access to mainframes
6. Each volume of hard disk can support different o/s format (e.g.. PC-DOS,Apple DOS,or CP/M)
7. Different computers an not however share single volume

ORIGINAL PAGE IS
OF POOR QUALITY

Davong Systems - Multilink

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	Multilink
2	Developer/Vendor:	Davong Systems
3	Address:	217 Humbolt Ct. Sunnyvale.CA 94089
4	Telephone No(s)/Contact:	(408) 734-4900
5	Documentation Availability/Source:	TBS
6	NOS Hardware Implementation:	
	---Network	Token ring
	---Host Computer(s)	IBM PC's (also Columbia.AT&T.Compaq.and ITT) Support for workstations from: ...Nestar.Datapoint.& Standard Microsystems
7	Host Operating System Name(s):	PC-DOS.MS-DOS
8	Implementation Language:	TBS
9	Maintenance Support:	On-site provided by RCA Carry-in service by Sorbus
10	ISO Model Implementation:	TBS
11	List of Services for Each ISO Layer:	File sharing, printer spooling
12	Major O/S Functions Implemented:	TBS
13	O/S Modification(s):	TBS
14	Price:	Workstation at $700 Starter kit at $2395
15	User Added Applications S/W:	

Comments:

1. Must have at least one PC on the network as file server and network controller
2. File locking via module SHRDOS
3. Supports up to 255 stations and 20.000' cable
4. Token-passing scheme based on ARCnet philosophy
5. Ring speeds at 2.5Mbps

Digital Equipment Corporation - DECnet

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	DECnet
2	Developer/Vendor:	Digital Equipment Corporation
3	Address:	8301 Professional Place Landover, MD 20785
4	Telephone No(s)/Contact:	(301) 459-7900 Gary Brown - Sales Rep.
5	Documentation Availability/Source:	Handbook: "Digital Networks - An Architecture with a Future" Software Product Description Contact DEC for comprehensive literature list Source code is available at no charge - users apply for it and agree not to disseminate it to others
6	NOS Hardware Implementation:	
	---Network	Ethernet,X.25,DDCMP
	---Host Computer(s)	DEC VAX,PDP,MicroVAX,DECSystem10,DECSystem20, Professional 300,Rainbow
7	Host Operating System Name(s):	VMS,RSX,RT-11,TOP-10,TOP-20,MicroVMS
8	Implementation Language:	TBS
9	Maintenance Support:	Software update service; Comprehensive H/W field engineering support
10	ISO Model Implementation:	Proprietary Digital Network Architecture (DNA) Model similar to 7 layer ISO model
11	List of Services for Each ISO Layer:	

Layer	DNA Label	Function
7	User/Net. Mngmt.	
6	Net. Applic.	Remote file access File transfer Remote terminal and resource access Network virtual terminal X.25 access SNA gateway access Remote command file submission
5	Session Control	Program - to - Program comms.
4	End-to-End Comms.	Assures against lost,duplicated packets
3	Routing	Best effort basis packet delivery
2	Data Link	X.25,Ethernet,DDCMP
1	Physical	"

ITEM #	ITEM	DESCRIPTION
12	Major O/S Functions Implemented:	Task-to-task communications,remote file and record access,file transfer,terminal communications, network virtual terminal
13	O/S Modification(s):	Runs under local o/s (e.g., VMS). DECnet in a UNIX environment available CY1985
14	Price:	Each node licensed based on CPU type (e.g., VAX 11/750, 11/780) cost is $1050 for DECnet license plus $630 for software
15	User Added Applications S/W:	Done via VMS devleopment facilities

Comments:

1. DNA nodes inreased from 255 to 1023
2. DECnet/SNA gateway available
3. Phase IV development in progress with current support for VAX systems
4. X.25 gateway available

ORIGINAL PAGE IS
OF POOR QUALITY

Digital Research Inc - DR NET

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	DR NET
2	Developer/Vendor:	Digital Research Inc
3	Address:	110 South Jefferson Rd Whippany, NJ 07981
4	Telephone No(s)/Contact:	(201) 428-1900
5	Documentation Availability/Source:	TBS
6	NOS Hardware Implementation:	
	---Network	TBS
	---Host Computer(s)	8086/8088-based computers
7	Host Operating System Name(s):	CP/M, PC-DOS and DR's Concurrent DOS
8	Implementation Language:	TBS
9	Maintenance Support:	TBS
10	ISO Model Implementation:	vendor claims "product lends itself to be built along OSI guidelines"
11	List of Services for Each ISO Layer:	
12	Major O/S Functions Implemented:	Remote file access (with file and record locking) Remote printer access
13	O/S Modification(s):	No modification to DR's Concurrent DOS Modification to other o/s is TBS
14	Price:	TBS
15	User Added Applications S/W:	TBS

Comments:

1. Offered as independent module for latest release of DR Concurrent DOS o/s
2. OEM's responsible for NIOS I/O to server and end-to-end provision of reliable virtual circuit service
3. Password protection on servers and private disk drives supported
4. NDOS acts as a redirector to route local vs remote o/s calls

Fox Research - 10-NET

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	10-NET
2	Developer/Vendor:	Fox Research, Inc
3	Address:	7005 Corporate Way Dayton, OH 45459
4	Telephone No(s)/Contact:	(513) 433-2238
5	Documentation Availability/Source:	TBS
6	NOS Hardware Implementation:	
	---Network	Ethernet
	---Host Computer(s)	PC,XT,AT
7	Host Operating System Name(s):	PC-DOS
8	Implementation Language:	TBS
9	Maintenance Support:	TBS
10	ISO Model Implementation:	TBS
11	List of Services for Each ISO Layer:	File server Electronic News Electronic Mail
12	Major O/S Functions Implemented:	TBS
13	O/S Modification(s):	Extensions to PC-DOS
14	Price:	TBS
15	User Added Applications S/W:	TBS

Comments: 1. Unsucessful reaching vendor via phone calls

ORIGINAL PAGE IS OF POOR QUALITY

Intel Corporation - OpenNet

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	OpenNet
2	Developer/Vendor:	Intel Corporation
3	Address:	3065 Bowers Ave. Santa Clara, CA 95051
4	Telephone No(s)/Contact:	(301) 796-7500 John Hollis, Chuck McKinney (MD ofc.)
5	Documentation Availability/Source:	Source Code available
6	NOS Hardware Implementation:	
	---Network	Ethernet (IEEE 802.3)
	---Host Computer(s)	Intel i310(8086,80286) IBM PC
7	Host Operating System Name(s):	iRMX, Xenix,PC-DOS,MS-DOS
8	Implementation Language:	"C"(Xenix version) PLM(iRMX version)
9	Maintenance Support:	Software Update Service
10	ISO Model Implementation:	Implements ISO/OSI Model
11	List of Services for Each ISO Layer:	L1,L2 - Ethernet L3 - Null L4 - ISO Class 4 Transport L5 - Intel Session protocoll L6 - Null L7 - Intel network file access access protocol
12	Major O/S Functions Implemented:	Electronic Mail (XENIX) Remote execution of batch jobs (XENIX) Remote print service Record locking File sharing,file access Hierarchical file structure
13	O/S Modification(s):	no modifications
14	Price:	TBS
15	User Added Applications S/W:	via iRMX or XENIX development s/w

Comments:

1. L1 - L4 of OSI model implemented in VLSI on a single board:
   - Intel SXM 552 transport engine
   - 186/51 COMMputer board
   - iNA960 transport layer s/w
   - 82586 LAN coprocessor
   - 82501 Ethernet controller
2. L5 - L7 support transparent interoperability between PC-DOS, MS-DOS, XENIX, & iRMX-86
3. Compatible with IBM-PC networking software and Microsoft MS-NET product
4. OpenNet to be implemented in DEC VAX/VMS environment by 4 QTR 1985

Lantech Systems, Inc. - uNETix

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	uNETix
2	Developer/Vendor:	Lantech Systems, Inc.
3	Address:	9635 Wendell Road Dallas, TX 75243
4	Telephone No(s)/Contact:	(214) 340-4932 Paul Greusel
5	Documentation Availability/Source:	TBS
6	NOS Hardware Implementation:	
	---Network	Ethernet,Corvus,Omninet and Percom LANs supported
	---Host Computer(s)	IBM PC and compatibles
7	Host Operating System Name(s):	UNIX V7 and System III compatibility
8	Implementation Language:	"C"
9	Maintenance Support:	TBS
10	ISO Model Implementation:	Not implemented
11	List of Services for Each ISO Layer:	N/A
12	Major O/S Functions Implemented:	Virtual Terminal Remote File and Device access Virtual File System IPC
13	O/S Modification(s):	uNETix kernal is proprietary implementation
14	Price:	$300 to $600 per node
15	User Added Applications S/W:	Software development tools available under uNETix

Comments:

1. Dynamic load balancing to be offered
2. System administrator provides audit trail of user accessed files
3. Compatible with UNIX file system
4. Up to 10 user defined windows

ORIGINAL PAGE IS
OF POOR QUALITY

Locus Computing Corporation - LOCUS

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	LOCUS
2	Developer/Vendor:	Locus Computing Corporation
3	Address:	3330 Ocean Park Blvd Santa Monica, CA 90405
4	Telephone No(s)/Contact:	Judi Uttal, Mktg. Mngr. (213) 452-2435
5	Documentation Availability/Source:	Source code prices: --OEM license $150,000 --Subsequent licenses same organization $10,000 --Right to provide binary sublicenses $25,000 "The LOCUS Distributed System Architecture" Edition 3.1 June 1984 Programmers Guide Enhancements to UNIX
6	NOS Hardware Implementation:	
	---Network	Ethernet (can also coexist w/XNS and ISO)
	---Host Computer(s)	DEC VAX,Motorola MC68000
7	Host Operating System Name(s):	Locus distributed UNIX ...object code compatibility with: - Berkeley UNIX (V4.1 and 4.2) - UNIX System V ...high degree of compatibility with others
8	Implementation Language:	"C"
9	Maintenance Support:	new releases of software h/w to OEM's only - no maint. avail.
10	ISO Model Implementation:	not implemented - uses proprietary protocol
11	List of Services for Each ISO Layer:	Vitually all services supported including: File Service........... Print Service........... Message Handling........ Remote Procedure Call... Job Transfer & Manip.... Terminal Handling....... Directory Service....... Data Representation..... Interprocess Comms...... Transport Services...... Recovery Services....... Network Management...... Network Security........
12	Major O/S Functions Implemented:	Load leveling Remote file access
13	O/S Modification(s):	A six step procedure (LOCUS Architecture Manual pp 56-67) is used to convert a UNIX system to a LOCUS system. This is done by replacing the UNIX kernal and associated system software (all the root file system). LOCUS also requires more disk space than conventional UNIX.
14	Price:	$500,000 to port LOCUS
15	User Added Applications S/W:	Easily done in UNIX environment

Comments:

1. LOCUS is sold primarily to OEM's
2. TCP/IP to be supported in near future
3. No ISO/OSI support envisioned for near future due to extensive

efforts to produce existing product
4. Customer must have an appropriate UNIX
license to run LOCUS distributed UNIX

Microsoft Corooration - MS-DOS & Networks

ITEM #	ITEM	DESCRIPTION
1	Name/Availabilitv of NOS:	Networks
2	Develooer/Vendor:	Microsoft Corooration
3	Address:	3075 112 Ave. NE Bellvue. WA 98004
4	Teleohone No(s)/Contact:	(206) 828-8080 (sales) (206) 826-8089 (tech support) 1-800-426-9400 (Bellvue)
5	Documentation Availabilitv/Source:	"Guidelines for Networkinq Product Develooment" "MS-DOS 3.1 Proorammers Guide" "Microsoft Networks Users Guide" "Microsoft Networks Manaoers Guide" "Server/Redirector File Sharainq Protocol" "Transport Laver Interface Guide" "IBM PC Compatibilitv"
6	NOS Hardware Imolementation:	
	---Network	Various
	---Host Comouter(s)	IBM-PC's
7	Host Ooerating Svstem Name(s):	MS-DOS 3.1
8	Imolementation Lanquage:	TBS
9	Maintenance Support:	TBS
10	ISO Model Imolementation:	Implements ISO L5.6.7
11	List of Services for Each ISO Laver:	L7 - File and orint server L6 - Redirector L5 - MS-DOS 3.1 L4.L3 - Network device driver s/w L3.L2.L1 - Network hardware
12	Major O/S Functions Imolemented:	TBD
13	O/S Modification(s):	TBD
14	Price:	TBD
15	User Added Aoolications S/W:	TBD

Comments:

1. Requires followinq:
   - -192KB RAM for workstations and server
   - -MS-DOS 3.1
   - -Networks 1.0
   - -LAN card suoolied bv other vendors
   - -Sinqle floppv drive (minimum)
   - -Transport interface card
2. Hardware indeoendent
3. Password orotection for directorv or subdirectorv
4. Read onlv/write only/create (or combination) oermission oranted
5. Anv block of characters in afile mav be locked
6. File sharinq.peripheral sharinq (e.g., printer)
7. Hiqh level orotocol between server and aoolications
8. Server commands:SHARE.PRINT.DEBUG.STATUS.STOP.HELP

Multi Solutions, Inc - S1

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	S1
2	Developer/Vendor:	Multi Solutions Inc
3	Address:	123 Franklin Corner Rd Suite 207 Lawrenceville, NJ 08648
4	Telephone No(s)/Contact:	Patricia McMahon (609) 896-4100
5	Documentation Availability/Source:	Source code not available
6	NOS Hardware Implementation:	
	---Network	TBS
	---Host Computer(s)	Z80,68000,others
7	Host Operating System Name(s):	S1
8	Implementation Language:	Pascal-like
9	Maintenance Support:	TBS
10	ISO Model Implementation:	Not implemented
11	List of Services for Each ISO Layer:	TBS
12	Major O/S Functions Implemented:	Electronic Mail File and Device sharing Intertask communications
13	O/S Modification(s):	proprietary o/s
14	Price:	TBS
15	User Added Applications S/W:	via S1 utilities (editor,compiler)

Comments:

1. Suited for real-time applications via priority scheduler and contiguous file support
2. Task wake-up on events

Nestar Systems Inc. - Plan 3000

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	Plan 3000,4000
2	Developer/Vendor:	Nestar Systems Inc.
3	Address:	2585 E. Bayshore Rd. Palo Alto, CA 94303
4	Telephone No(s)/Contact:	(415) 493-2223 (202) 289-1672 - Craig Marr or J bullard (703) 968-6554 - J. Bullard
5	Documentation Availability/Source:	Source code not available Other documentation TBD
6	NOS Hardware Implementation:	
	---Network	Arbitrary tree,baseband,token passing
	---Host Computer(s)	IBM PC's, Apples(primarily as print servers)
7	Host Operating System Name(s):	DOS 2.1,3.0
8	Implementation Language:	Pascal
9	Maintenance Support:	h/w - RCA contract s/w - Nestar maintenance contracts
10	ISO Model Implementation:	Conforms to ISO/OSI model (no add'l info)
11	List of Services for Each ISO Layer:	Shared disks,print servers Electronic mail,chat,word processor spreadsheet,accounting program and multi-user database
12	Major O/S Functions Implemented:	Password access security Log-on ID security File,record protection
13	O/S Modification(s):	None to DOS on PC's
14	Price:	Workstation $600 Dedicated server $12,995
15	User Added Applications S/W:	Possible

Comments:

1. Dedicated file servers of up to 56MB each
2. Can network up to 255 stations
3. Fibre optic cabling supported
4. A workstation can mount up to 26 drives at one time
5. Public,group,and private controlled access to virtual disks
6. Any PC can be designated as a print server
7. SNA gateway available

Network Research Corporation - FUSION

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	FUSION
2	Developer/Vendor:	Network Research Corporation
3	Address:	1964 Westwood Blvd. Suite 200 Los Angeles, CA 90025
4	Telephone No(s)/Contact:	213-474-7717 525-4141/Lee Cowan
5	Documentation Availability/Source:	Source code available - contact vendor List of available documentation - TBS
6	NOS Hardware Implementation: ---Network ---Host Computer(s)	 Ethernet IBM-PC,MC68000,VAX,PDP-11,Intel 8086,8088. NS 16032, F-11
7	Host Operating System Name(s):	UNIX ---Berkeley 4.1,4.2 ---Version 7 ---System 3 ---System 5 ---Venix ---Xenix ---Ultrix MS-DOS 2.0 PC-DOS VMS
8	Implementation Language:	"C"
9	Maintenance Support:	Installation and maintenance provided
10	ISO Model Implementation:	Provides user with functionality of ISO/OSI layers 3-7. Supports Xerox XNS: TCP/IP protocols.
11	List of Services for Each ISO Layer:	Layers 1,2: Ethernet H/W interface Layers 3,4: FUSION Socket Manager Layers 5,6,7: Remote execution and virtual terminal
12	Major O/S Functions Implemented:	File Transfer Virtual Terminal Remote Execution Network Utilities:traffic monitoring; file,mail,and print servers;performance analysis
13	O/S Modification(s):	none - FUSION Socket Manager is integrated into the kernal as a device driver
14	Price:	$750 - $8000 (CPU & O/S dependent)
15	User Added Applications S/W:	Add to library routines - compatible system calls

Comments:

(1) File Transfer - A single file or complete directories of files can be transferred between a wide variety of different computers and operating systems.

(2) Virtual Terminal - Each computer can act as a terminal to any other computer on the FUSION network.

(3) Remote Execution - Commands may be executed on a remote computer having a different operating system,different processor,or different network hardware.

(4) The FUSION Socket Manager (ISO L3-4) requires approximately 18KB of program space and 10-20KB buffer space. PC's require about 13KB program space.

(5) 1Mbps throughput rates on a virtual circuit. application-to-application connections (i.e.. ISO L7 to L7).

Network Systems Corporation - NETEX

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	NETEX
2	Developer/Vendor:	Network Systems Corporation
3	Address:	7600 Boone Avenue North Minneapolis, MN 55428-1099
4	Telephone No(s)/Contact:	(612) 425-2202 Bob Smyth
5	Documentation Availability/Source:	"Network Executive (NETEX) Software and NETEX Utilities" --- Add'l documentation TBS --- Source code available only to existing customers
6	NOS Hardware Implementation: ---Network ---Host Computer(s)	 HYPERchannel IBM,DEC,Univac, & CDC
7	Host Operating System Name(s):	IBM MVS/SP3 & VM/SP DEC PDP-11 RSX-11M/M+ Release 3.2 DEC VAX VMS Release 2.2 Univac OS1100 CDC NOS & NOS/BE
8	Implementation Language:	Pascal
9	Maintenance Support:	Extensive field engineerin for h/w Software tape update service ala IBM
10	ISO Model Implementation:	Implements ISO 7 layer model
11	List of Services for Each ISO Layer:	L1,2 - HYPERchannel L3,4,5 - NETEX and host O/S (interprocess comms) L6,7 - User written s/w and utilities e.g., NSC Bulk File Transfer(BFX) and and Data Representation Service
12	Major O/S Functions Implemented:	L7 - User written s/w L6 - User s/w and NSC utilities L5 - program-to-program connection - read/write data - disconnection - statistics gathering L4 - data delivery assurance - buffering for FDX communications L3 - driver and host sublayers -Driver Sublayer
13	O/S Modification(s):	NETEX implemented as a separate program under the local o/s control and called by applications programs.
14	Price:	About \$1300 - \$6000 (depending upon implementation and specific system)
15	User Added Applications S/W:	NSC permits user's to write Applications and Presentation layer programs using NETEX

Comments:

1. NETEX may be used in a heterogeneous computer environment provided each computer is provided with HYPERchannel interface and proper version of NETEX
2. Handles delay compensation for T1 satellite communications links
3. Does not support peripheral devices attached to the HYPERchannel: it is for host-to-host comms.
4. Not transparent to the user. It must be called by an

application or utility to use its services

5. Security relies on host o/s implementation features
6. BFX utility efficient for files over 1MB
7. Future development efforts:
   -NETEX for Apollo computers
   -NETEX on boards for Multibus and PC-bus machines

Novell Inc. - NETWARE

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	NETWARE (available since 1983)
2	Developer/Vendor:	Novell.Inc.
3	Address:	1170 N. Industrial Park Drive Orem. UT 84057
4	Telephone No(s)/Contact:	1-800-453-1267 Robert Walton
5	Documentation Availability/Source:	Systems Guide Programmers Guide Technical Reference Manual
6	NOS Hardware Implementation:	
	---Network	Star network w/ twisted pair Compatible with: 3COM Etherlink. Nestar PLAN 2000.Orchid's PCNet. IBM PC Cluster. Davong MULTILINK. and Proteon PRONET.
	---Host Computer(s)	IBM PC's and plug compatibles. Many MS-DOS computers.
7	Host Operating System Name(s):	PC-DOS.MS-DOS
8	Implementation Language:	Lattice "C"
9	Maintenance Support:	Hardware through distributors/dealers Software through update service
10	ISO Model Implementation:	Implements ISO model
11	List of Services for Each ISO Layer:	TBS
12	Major O/S Functions Implemented:	File server.Print spooler. and Electronic mail
13	O/S Modification(s):	Network interface shell over PC-DOS for interpreting network commands'network commands Server has multiuser.multitasking o/s
14	Price:	$1495 for NETWARE 10.E.P (ie: Corvus.3COM.and Proteon look-alikes) $795 for Electronic Mail Hardware price is a function of the network hardware selected
15	User Added Applications S/W:	Suppoted via Programmers Reference Guide and Technical Reference Manual

Comments:

(1) Protocol is a proprietary.polled differential line
(2) Bandwidth at baseband is 600 kbps
(3) Proprietary server based on 68000 processor
(4) Security - limited access to subdirectories and volumes: file and record locking: name and password logon
(5) Up to 5 printers on server: print spooling
(6) Runs DOS 3.00 and 3.10 S/W
(7) Additional function numbers have been added to DOS and defined for Netware to allow operations not supported in native O/S (e.g.. File Lock).

Orchid Technology - PCnet

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	PCnet
2	Developer/Vendor:	Orchid Tecnology
3	Address:	47790 Westinghouse Drive Freemont, CA 94539
4	Telephone No(s)/Contact:	(415) 490-8586
5	Documentation Availability/Source:	TBS
6	NOS Hardware Implementation:	
	---Network	CSMA/CD coax cable
	---Host Computer(s)	IBM PC's, XT's
7	Host Operating System Name(s):	PC-DOS, MS-DOS
8	Implementation Language:	TBS
9	Maintenance Support:	TBS
10	ISO Model Implementation:	TBS
11	List of Services for Each ISO Layer:	File and device sharing (e.g., modems,disks,printers)
12	Major O/S Functions Implemented:	File locking Limited RPC Messaging
13	O/S Modification(s):	TBS
14	Price:	Workstation $495 Starter kit $1090
15	User Added Applications S/W:	TBS

Comments:

1. Up to 256 PC's supported
2. 1Mbps network data rates via CSMA/CD
3. Each user workstation can access up to 14 servers
4. Any PC in the system can function as server or workstation

Plexus Computers - NOS

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	NOS
2	Developer/Vendor:	Plexus Computers
3	Address:	3833 N. First St. San Jose, CA 95134
4	Telephone No(s)/Contact:	(408) 943-9433 California (201) 843-7766 New Jersey
5	Documentation Availability/Source:	TBS - telecon 4/29
6	NOS Hardware Implementation:	TBS - telecon 4/29
	---Network	TBS - telecon 4/29
	---Host Computer(s)	TBS - telecon 4/29
7	Host Operating System Name(s):	TBS - telecon 4/29
8	Implementation Language:	TBS - telecon 4/29
9	Maintenance Support:	TBS - telecon 4/29
10	ISO Model Implementation:	TBS - telecon 4/29
11	List of Services for Each ISO Layer:	TBS - telecon 4/29
12	Major O/S Functions Implemented:	TBS - telecon 4/29
13	O/S Modification(s):	TBS - telecon 4/29
14	Price:	TBS - telecon 4/29
15	User Added Applications S/W:	TBS - telecon 4/29

Comments: TBS - telecon 4/29

ORIGINAL PAGE IS
OF POOR QUALITY

Portable Software Inc. - The Newcastle Connection

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	The Newcastle Connection
2	Developer/Vendor:	Portable Software Inc
3	Address:	650 Bair Island Road Suite 204 Redwood City, CA 94063
4	Telephone No(s)/Contact:	Keith Clark (415) 367-6264
5	Documentation Availability/Source:	Source Code Available Manuals: Installation Procedures Interfacing to the Network The RPC Protocol
6	NOS Hardware Implementation:	
	---Network	Ethernet,Omninet,Cambrige Token Ring Net
	---Host Computer(s)	AT&T 3B,68000-based(Corvus,Sun),PDP-11 VAX,Pyramid,Gould SEL and custom bit slice processor
7	Host Operating System Name(s):	UNIX System V, System III, V7 4.2BSD and XENIX
8	Implementation Language:	"C"
9	Maintenance Support:	Software Update Service
10	ISO Model Implementation:	Applications Layer
11	List of Services for Each ISO Layer:	Electronic mail File Service
12	Major O/S Functions Implemented:	Remote Procedure Call Interprocess Communication
13	O/S Modification(s):	None
14	Price:	OEM binary licence per node: $750 Source Code license: $3000
15	User Added Applications S/W:	Yes - via host o/s software development packages

Comments:

1. TNC uses 'redirector' to send calls out to the network instead of local o/s when required

Quadram Corporation - Quadnet VI

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	Quadnet VI
2	Developer/Vendor:	Quadram Corporation
3	Address:	4357 Park Drive Norcross, GA 30093
4	Telephone No(s)/Contact:	(404) 923-6666
5	Documentation Availability/Source:	TBS
6	NOS Hardware Implementation:	
	---Network	CSMA (with both CD and CA)
	---Host Computer(s)	IBM PC's, XT's, or compatibles
7	Host Operating System Name(s):	PC-DOS, MS-DOS
8	Implementation Language:	TBS
9	Maintenance Support:	Limited diagnostics,return shipping for repairs. Telephone tech assistance
10	ISO Model Implementation:	TBS
11	List of Services for Each ISO Layer:	Electronic mail,word processing Shared devices (printers,disks,plotters) Print spooling
12	Major O/S Functions Implemented:	Log-on ID security File,record protection
13	O/S Modification(s):	TBS
14	Price:	Workstation $595 Server station $1295
15	User Added Applications S/W:	TBS

Comments:

1. Network uses both CA and CD aspects of CSMA protocol. Using both increases reliability but decreases s/w overhead.
2. Up to 255 workstations; 1 server
3. Server is a dedicated XT
4. Workstation devices cannot be shared but can only be used locally.
5. Uses Novell Netware software
6. Applications designed to operate on CP/M 86, Concurrent DOS, UNIX, and Xenix are not usable.
7. Hard disk server is public (SYS:PUBLIC) - user volumes are DOS subdirectories.

Quadram Corporation - Quadnet IX

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	Quadnet IX
2	Developer/Vendor:	Quadram Corporation
3	Address:	4357 Park Drive Norcross, GA 30093
4	Telephone No(s)/Contact:	(404) 923-6666
5	Documentation Availability/Source:	TBS
6	NOS Hardware Implementation:	
	---Network	Token star-ring,10Mbps w/proprietary h/w twisted pair or fibre optics
	---Host Computer(s)	IBM PC's, XT's, or compatibles
7	Host Operating System Name(s):	PC-DOS, MS-DOS
8	Implementation Language:	TBS
9	Maintenance Support:	Limited diagnostics,return shipping for repairs. Telephone tech assistance
10	ISO Model Implementation:	TBS
11	List of Services for Each ISO Layer:	Electronic mail(EMS),word processing Shared devices (printers,disks,plotters) Print spooling
12	Major O/S Functions Implemented:	Log-on ID security File,record protection
13	O/S Modification(s):	TBS
14	Price:	Workstation $795 Server station $1495 4-station connector $195 4-station config. $9265
15	User Added Applications S/W:	TBS

Comments:

1. Uses Novell Netware software
2. Apparently uses proprietary h/w from Proteon
3. 4,8, or 16 workstations per ring
4. Up to 255 nodes
5. Interfaces available for:
   - -DEC Unibus (PDP & VAX)
   - -DEC Q-bus
   - -Intel Multibus
6. Use of Novell s/w means lack of compatibility with standard DOS device drivers
7. Proprietary Novell file server
8. 1 XT file server

Quantum Software Systems Ltd. - QNX

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	QNX 2.0
2	Developer/Vendor:	Quantum Software Systems Ltd.
3	Address:	Moodie Drive High Tech Park 215 Stafford Road, Unit 104 Ottawa, Ontario K2H 9C1 Canada
4	Telephone No(s)/Contact:	(613) 726-1893 Sandy Kingston
5	Documentation Availability/Source:	Source code not avil. Other doc. TBS
6	NOS Hardware Implementation:	
	---Network	Arcnet
	---Host Computer(s)	IBM PC,AT and compatibles also other 8088,8086,80186, & 80286 cpu's
7	Host Operating System Name(s):	QNIX (UNIX derivative)
8	Implementation Language:	"C"
9	Maintenance Support:	h/w - board exchange policy s/w - telephone technical support - on-line bulletin board service w/down-line load of new s/w revs.
10	ISO Model Implementation:	TBS (Unknown)
11	List of Services for Each ISO Layer:	TBS
12	Major O/S Functions Implemented:	File Service...........Y Print Service..........Y Message Handling........Y Remote Procedure Call...Y Job Transfer & Manip....Y Terminal Handling.......Y Directory Service.......Y Data Representation.....Y Interprocess Comms......Y Transport Services......Y Recovery Services.......? Network Management......? Network Security........Y
13	O/S Modification(s):	Development of QNIX (UNIX like) o/s QNIX - multiuser,multitasking,real-time LAN o/s
14	Price:	1-4 nodes is $1300 (per node) 5 or more nodes $2600
15	User Added Applications S/W:	Possible in QNIX environment

Comments:

1. X.25 public network access
2. Supports distributed processing as well as distributed devices
3. Pure processing elements (i.e., computers without keyboards or displays can be plugged into the network
4. Support DOS emulator to enable PC DOS tasks under QNIX

Standard Microsvstems Corooration - Arcnet LAN

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	Arcnet LAN
2	Developer/Vendor:	Standard Microsvstems Corooration
3	Address:	35 Marcus Blvd. Hauppauge. NY 11788
4	Telephone No(s)/Contact:	(516) 273-3100 Jim Hall
5	Documentation Availabilitv/Source:	TBS
6	NOS Hardware Implementation:	
	---Network	Token-ring
	---Host Computer(s)	IBM-PC and S-100 bus systems
7	Host Operating System Name(s):	TBS
8	Implementation Language:	TBS
9	Maintenance Support:	TBS
10	ISO Model Implementation:	TBS
11	List of Services for Each ISO Layer:	TBS
12	Major O/S Functions Implemented:	TBS
13	O/S Modification(s):	TBS
14	Price:	TBS
15	User Added Applications S/W:	TBS

Comments:

1. Telecon w/ J Hall 4/24 - will send lit
didn't know what ISO was

Sun Microsystems' - NFS

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	NFS 2.0 (April 1985)
2	Developer/Vendor:	Sun-Microsystems Inc.
3	Address:	8233 Old Courthouse Road. Suite 200 Vienna, VA 22180
4	Telephone No(s)/Contact:	(703) 883-0444
5	Documentation Availability/Source:	"Overview of the Sun Network File System" "Remote Procedure Call Protocol Specification" "Remote Procedure Call Reference Manual" "External Data Representation Reference Manual" Source Code Available - $25,000
6	NOS Hardware Implementation:	
	---Network	Ethernet
	---Host Computer(s)	Sun Workstations,DEC/VAX,IBM-PC
7	Host Operating System Name(s):	UNIX (also non-UNIX environments) VAX/VMS. VAX/UNIX
8	Implementation Language:	TBD
9	Maintenance Support:	TBS
10	ISO Model Implementation:	Does not conform to ISO
11	List of Services for Each ISO Layer:	File sharing Remote Procedure Call
12	Major O/S Functions Implemented:	File sharing in an heterogeneous computer environment. Remote Procedure Call (with authentication)
13	O/S Modification(s):	Extension of UNIX O/S
14	Price:	$1200 per Sun Workstation
15	User Added Applications S/W:	TBS

Comments:

(1) The intent of NFS is to make it transmission technology and operating system independent.

(2) NFS provides an environment that can accomodate hardware from other vendors.

(3) Sun's implementation resides in the UNIX kernal (UNIX 4.2) and are proprietary.

(4) Yellow Pages will be another network service. It is a phone book of available network resources for clients.

(5) Remote copying rates are expected to be greater than 100 KBytes/sec.

(6) NFS increases UNIX kernal size 20-25%. They suggest 2MB main memory to minimize adverse effects on performance.

(7) File locking capability is expected future enhancement.

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	PM/16 (since June 1984)
2	Developer/Vendor:	TeleVideo Systems. Inc.
3	Address:	1170 Morse Ave. Sunnyvale. CA 94086
4	Telephone No(s)/Contact:	(408) 745-7760 (408) 471-0255
5	Documentation Availability/Source:	400 page programmers manual to be released
6	NOS Hardware Implementation: ---Network ---Host Computer(s)	 IBM PC's and compatibles
7	Host Operating System Name(s):	MS-DOS, PC-DOS V1.1. V2.x Note: V3.0 and 3.1 avail. soon
9	Maintenance Support:	6-month warranty TRW & Xerox offer on-site/depot service
10	ISO Model Implementation:	TBS
11	List of Services for Each ISO Layer:	Shared printers.disks.plotters,modems User-User messaging Network Management (see comments)
12	Major O/S Functions Implemented:	Security: -file level r/w.open.create,delete access permission -directory search permission -logon passwords
13	O/S Modification(s):	TBS
14	Price:	About $2500 per node
15	User Added Applications S/W:	TBS

Comments:

1. Infoshare is TeleVideo custom version of Novell NetWare
2. Uses dedicated file server connected in star topology with workstations
3. Up to 16 workstations may be connected to server
4. Fibre optic modem available
5. 3270 SNA gateway available
6. Network manager can:
   - -broadcast messages to all users logged on
   - -log stations off
   - -prevent add'l. log-ons
   - -control print spooling (e.g., reroute print tasks)
   - -perform overall network usage monitoring
7. Electronic mail not currently avail.

3Com Corporation - Etherseries

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	EtherSeries
2	Developer/Vendor:	3Com Corporation
3	Address:	1365 Shore Bird Way Mountainview. CA 94043
4	Telephone No(s)/Contact:	(415) 961-9602 Jeff Perez
5	Documentation Availability/Source:	Source Code negotiated individually User Software Guide Network Administrators Guide Internals software guide
6	NOS Hardware Implementation: ---Network ---Host Computer(s)	 Ethernet (IEEE 802 CSMA/CD) IBM PC,XT.AT
7	Host Operating System Name(s):	PC-DOS
8	Implementation Language:	
9	Maintenance Support:	Software Updates provided 5 day turnaround on boards XEROX Americare for servicing
10	ISO Model Implementation:	L1.L2: Ethernet L3 - L5: XNS protocol L6.L7:Proprietary
11	List of Services for Each ISO Layer:	Electronic mail Disk.printer sharing
12	Major O/S Functions Implemented:	TBS
13	O/S Modification(s):	The only departure from standard PC-DOS usre interface are commands and utilities to support multiple users in a networked environment.
14	Price:	$650 and up
15	User Added Applications S/W:	Possible - use internals software guide

Comments:

Touchstone Software Corporation - The Connectables Network

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	The Connectables Network ---PCworks for PC's ---UniHost for UNIX,Xenix systems ---MacLine for Macintosh
2	Developer/Vendor:	Touchstone Software Corporation
3	Address:	909 Electric Avenue Seal Beach,CA 90740
4	Telephone No(s)/Contact:	(213) 598-7746 Stephen Onstott Corporate Account Representative
5	Documentation Availability/Source:	Source code not available Manuals for PCworks,UniHost,MacLine
6	NOS Hardware Implementation: ---Network ---Host Computer(s)	 Dial-up, proprietary protocol PC's,UNIX systems, & Macintosh
7	Host Operating System Name(s):	UNIX,PC-DOS
8	Implementation Language:	"C"
9	Maintenance Support:	No custom hardware required Software update sevice charge (~$30)
10	ISO Model Implementation:	Not implemented
11	List of Services for Each ISO Layer:	File transfer Electronic Mail Remote terminal Limited shared devices Status reporting Remote printing
12	Major O/S Functions Implemented:	File Service...........Y Print Service..........Y Message Handling........Y Remote Procedure Call...N Job Transfer & Manip....N Terminal Handling.......Y Directory Service.......N Data Representation.....Y Interprocess Comms......N Transport Services......Y Recovery Services.......Y Network Management......Y Network Security........Y
13	O/S Modification(s):	none to PC-DOS UNIX mods unknown
14	Price:	UniHost license - $295 PCworks license - $195 MacLine license - $195
15	User Added Applications S/W:	Hooks are there for add-ons but only for internal use for now

Comments:

1. Remote terminal support for standard ANSI,TTY, or VT100/52
2. UniHost requires:
   - -UNIX o/s
   - -30-60 KB disk/memory space
   - -serial log-in port

-RS232C interface cable
3. PCworks requires:
   -128KB IBM/Comptible PC
   -DOS 2.0 Operating System
   -280 KB disk storage
   -Asynchronous adapter
   -RS232C interface cable
4. MacLine requires:
   -128KB Macintosh
   -RS232C serial interface cable
5. Proprietary communications orotocol

Ungermann-Bass - NET/ONE

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	NET/ONE Personal Connection (avail Fall 1984)
2	Developer/Vendor:	Ungermann-Bass Inc.
3	Address:	2560 Mission College Blvd. Santa Clara. CA 950509
4	Telephone No(s)/Contact:	(408) 496-0111 Pat Bolger
5	Documentation Availability/Source:	Source code avail. - negotiated price "NET/ONE O/S MANUAL" "NET/ONE MANAGEMENT CONTROL"
6	NOS Hardware Implementation:	
	---Network	Ethernet.twin-coax baseband (RG-58) Broadband.& Fibre Optic Star
	---Host Computer(s)	IBM PC and plug compatibles Machines with RS-232C.IEEE 488.or DEC DR-11-W/B ports.or 32 bit parallel I/F's
7	Host Operating System Name(s):	PC-DOS 2.0 or higher Microsoft Network (future)
8	Implementation Language:	TBS
9	Maintenance Support:	About 1/10'th s/w cost for annual updates Field engineers for on-site maint. Monthly maint. contracts avail.
10	ISO Model Implementation:	L1 - L3 only L4 -L7 proprietary
11	List of Services for Each ISO Layer:	High level applications:"Crosstalk".Word Processing. some Interprocess Comms.
12	Major O/S Functions Implemented:	Diskshare (not a file server) Printshare (Spooled printing) Mailshare (Electronic mail) X.25 Gateway (requires special H/W)
13	O/S Modification(s):	O/S mods made - details unavail.
14	Price:	$1095 for NIU board $1000 for Diskshare or Printshare $ TBD for Mailshare
15	User Added Applications S/W:	Is possible (e.g.. Lotus 123.WordStar)

Comments:

(1) Provides file locking
(2) Logon with name and password
(3) Bus and modified star with fibre optics available
(4) CSMA/CD or CSMA protocol
(5) PC XT or compatible server
(6) Any node can be used as a server and run applications at the same tyime
(7) Volumes may be private or shared: password volume protection available
(8) Any node can be used as a print server

ViaNetix, Inc - ViaNet

ORIGINAL PAGE IS
OF POOR QUALITY

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	ViaNet
2	Developer/Vendor:	ViaNetix,Inc
3	Address:	5766 Central Ave Boulder, CO 80301
4	Telephone No(s)/Contact:	(303) 440-0700 Howard Kelly, VP Business Operations John Ingals, West Coast Regional Sales Mgr.
5	Documentation Availability/Source:	Source code available - negotiable ViaNet General Information Manual ViaNet Intermediate Users Reference Manual ViaNet Network Administrators Guide
6	NOS Hardware Implementation:	
	---Network	Arcnet,Omninet,Ethernet,IBM networks
	---Host Computer(s)	IBM PC
7	Host Operating System Name(s):	MS-DOS,UNIX III, System V (future)
8	Implementation Language:	"C"
9	Maintenance Support:	TBS
10	ISO Model Implementation:	yes
11	List of Services for Each ISO Layer:	Presentation - XLAT which intercepts all system calls Session - Local Router (i.e., a redirector) Transport - Format Converter Out (converts system call to network message) Network - Network Router handles network I/O driver
12	Major O/S Functions Implemented:	File I/O Interprocess Communications via named pipes Directory service of users logged on
13	O/S Modification(s):	"minor modifications to O/S"
14	Price:	approx. $600 per node
15	User Added Applications S/W:	TBS

Comments:

1. File protection (read only, R/W file access)
2. Data,programs and devices are sharable
3. File and record locking
4. ViaNet is MS-DOS applications program in .EXE file format
5. All system calls pass through network software
6. Uses interrupt 21H for ViaNet communications
7. Uses redirector concept
8. UNIX and DOS can coexist with file transfer service
9. Originally done by XENIX and purchased in prototype software

Wang Laboratories - WangNet

ITEM #	ITEM	DESCRIPTION	
1	Name/Availability of NOS:	WangNet	
2	Developer/Vendor:	Wang Laboratories	
3	Address:		
		Lowell. Mass.	
4	Telephone No(s)/Contact:	1-800-225-0234	
		Mally <last name ?>	
5	Documentation Availability/Source:	TBS	
6	NOS Hardware Implementation:		
	---Network	Dual cable broadband.FDM and TDM	
	---Host Computer(s)	Wang VS computer	
7	Host Operating System Name(s):	TBS	
8	Implementation Language:	TBS	
9	Maintenance Support:	TBS	
10	ISO Model Implementation:	Uses compressed form to just 3 layers	
		ISO Layer(s)	Wang Name
		1 - 4	Transport
		5 - 6	Services
		7	Applications
11	List of Services for Each ISO Layer:	Transport:	X.25.SNA/BSC,PTP.WangBand
		Services:	File Transfer Manager
			Remote Operating System Access
			Wang VS Computer Terminal Emulation
			Virtual Terminal Interface
		Applications:	Wang Office Software
			Applications Services Architecture
			Wang Information Transfer Architecture, and
			Word Processing System - Communications Specific
12	Major O/S Functions Implemented:	Exchange files	
		Dumb Terminal Emulation	
		Program-to-Program comms between Wang VS hosts	
		Resource Sharing	
13	O/S Modification(s):	TBS	
14	Price:	TBS	
15	User Added Applications S/W:	TBS	

Comments:

Xcomp - X-NET

ITEM #	ITEM	DESCRIPTION
1	Name/Availability of NOS:	X-Net
2	Developer/Vendor:	Xcomp
3	Address:	4223 Ponderosa Ave. San Diego, CA 92123
4	Telephone No(s)/Contact:	(619) 573-0077
5	Documentation Availability/Source:	TBS
6	NOS Hardware Implementation:	
	---Network	bus architecture
	---Host Computer(s)	IBM PC's
7	Host Operating System Name(s):	MS-DOS 2.0
8	Implementation Language:	TBS
9	Maintenance Support:	TBS
10	ISO Model Implementation:	TBS
11	List of Services for Each ISO Layer:	Device sharing (disk,printer,any RS-232 device) RPC
12	Major O/S Functions Implemented:	Copies files
13	O/S Modification(s):	TBS
14	Price:	Workstation $395 Starter kit $995 Four station config. $6180
15	User Added Applications S/W:	TBS

Comments:

1. Comes with no applications software
2. Does not support:
   - electronic mail
   - chat between users
   - create print queue
3. No network status reporting

APPENDIX B

-

Bibliography of Relevant Literature

ORIGINAL PAGE IS OF POOR QUALITY

NASA/GSFC : Network Operating Systems Bibliography of Relevant Literature

Document #	Author(s)	Title
1	Ferrari,D.	The Evolution of Berkely UNIX
2	Gusela,R. & Zatti,S.	TEMPO - A Network Time Controller for a Distributed UNIX System
3	Jensen, E.D. & Pleszkoch,N.	ArchOS: A Physically Dispersed Operating System
4	Goscinski, A. & Indulska, J.	An Object Approach To Network Operating System Model Construction
5	McKendry,M.S.	Clouds: A Fault Tolerant Distributed Operating System
6	Shoch, J.F. & Hupp,J.A.	Notes on the "Worm" programs -- some early experience with distributed computation
7	Watson, R.W. & Fletcher,J.G.	An Architecture for Support of Network Operating System Services
8	Nessett,D.M.	A Systematic Methodology for Analyzing Security Threats to Interprocess Communication in a Distributed System
9	Zhongxiu, S.,Du Z, & Peigen Y.	ZCZOS: A Distributed Operating System for a LSI-11 Microcomputer Network
10	Sha L, Jensen, E.D., Rashid,R.F., and Northcutt J.D.	Distributed Co-operating Processes and Transactions
11	Battarel,G.J. & Savary,H.F.	Interprocess Communication System of the MT35 Digital Exchange
12	Tokuda,H. & Manning,E.	An Interprocess Communication Model for a Distributed Software Testbed
13	Liu,M.T. & Tsay,Duen-Ping	MIKE: A Network Operating System for the Distributed Double-Loop Computer Network
14	Copeland,J., Schnatmeier V., and Winston, A.	Locus: A Transparently Clear Solution
15	Winston,A.	The LOCUS PC Bridge
16	Brereton, P.	Detection and Resolution of Inconsistencies among Distributed Replicates of Files
17	Mamrak, S.A. Leinbaugh,D. and Berk, T.S.	A Progress Report on the Desperanto Research Project - Software Support for Distributed Processing
18	Peinl, P. and Reuter, A.	Synchronizing Multiple Database Processes in a Tightly Coupled Multiprocessor Environment
19	Chesley, H.R. and Hunt, V.B.	Squire - A Communications-Oriented Operating System
20	Foster, D.V., Dowdy, L.W., and Ames, J.E.	File Assignment in a Computer Network
21	Sollins, K.R.	Copying Structured Objects in a Distributed System
22	Stankovic, J.A.	A Perspective on Distributed Computer Systems
23	Spector, A.Z. and Schwarz, P.M.	Transactions: A Construct for Reliable Distributed Computing
24	- various -	- various -
25	Fair,E.	USENET: Spanning the Globe

ORIGINAL PAGE IS OF POOR QUALITY

NASA/GSFC : Network Operating Systems Bibliography of Relevant Literature

Document #	Source Publication	Date	Key Words/Phrases
1	IEEE:Distributed Processing Technical Committee Newsletter	June 1984	UNIX,distributed environment
2	IEEE:Distributed Processing Technical Committee Newsletter	June 1984	UNIX,synchronization
3	IEEE:Distributed Processing Technical Committee Newsletter	June 1984	Decentralized
4	IEEE:Distributed Processing Technical Committee Newsletter	June 1984	NOS,distributed resources
5	IEEE:Distributed Processing Technical Committee Newsletter	June 1984	Fault-tolerance,distributed O/S
6	XEROX Palo Alto Research Center	September 1980	Distributed computing,Ethernet
7	Computer Networks 4	April(?) 1980	NOS,network architecture,distributed computing
8	IEEE Transactions on Communications	September 1983	Network security,distributed system, interprocess communication
9	ACM Operating Systems Review	July 1983	
10	ACM Computer Communications Review Vol 13, No 2 - SIGCOM '83 Symposium	March 1983	Distributed system,synchronization,NOS
11	ACM Computer Communications Review Vol 13, No 2 - SIGCOM '83 Symposium	March 1983	Interprocess communication,fault tolerance,ISO layers
12	ACM Computer Communications Review Vol 13, No 2 - SIGCOM '83 Symposium	March 1983	Distributed software,interprocess communication
13	IEEE Transactions on Software Engineering	March 1983	NOS,distributed systems,computer networks, resource sharing,synchronization
14	UNIX/World	Vol 1, No 6 1984	UNIX,Ethernet,synchronization,reliability
15	UNIX/World	Vol 1, No 6 1984	UNIX,LAN,IBM-PC,Ethernet
16	ACM Operating Systems Review	January 1983	NOS,file access,file replication
17	ACM Operating Systems Review	January 1983	Distributed computing,resource sharing, networking
18	ACM Operating Systems Review	January 1983	Database systems,synchronization
19	Computer Networks	September 1981	Real-time operating system, multitasking, microprocessors,distributed processors
20	Computer Networks	September 1981	NOS,distributed databases,computer networks
21	Computer Networks	September 1981	Distributed system
22	IEEE Transactions on Computers	December 1984	Communications subnet,computer networks, distributed computer system,distributed databases,distributed operating systems, distributed processing
23	ACM - Operating Systems Review	April 1983	Distributed computing,synchronization, communication
24	ACM - Operating Systems Review Special Issue - Proceedings of the Ninth Principles 10-13 October 1983	Vol 17, No. 5	Distributed computing,network communication
25	UNIX/World	Vol 1, No 7, 1984	UNIX,network communications,broadcast network

NASA/GSFC : Network Operating Systems Bibliography of Relevant Literature

Document #	Author(s)	Title
26	Jensen, E.D.	Physically Dispersing an Operating System
27	McGann, L.B.	Courier Protocol Accesses Services with Remote Procedure Calls
28	White, J.E.	Distributed, Replicated Software Locates Network Resources
29	Nelson, D.L.	Virtual Memory/Addressing Manages Token-Passing Ring's Resources
30	Jarvis,P.	Disk Emulation Allows Ethernet Resource Sharing
31	Ulrich, W.E.	Computer-Based Message Systems: Centralized or Distributed ?
32	Ennis, G.	Routing Tables Locate Resources in Bridged Broadband Networks
33	Wesel, G.W.	Transport Layer Utility Program Supports Network Management
34	Davidson, J.M.	User-Programmable Local Network Handles Seven Protocol Layers
35	Rauch-Hindin, W.B.	Netway Connects Diverse Equipment and Protocols
36	Rauch-Hindin, W.B.	Upper-Level Network Protocols
37	Durr,Michael	Advanced Netware
38	Walker B, Popek G, English R, Kline C, and Thiel G	The LOCUS Distributed Operating System
39	Tanenbaum A,Mullender S	An Overview of the Amoeba Distributed Operating System
40		Wang's Broad Approach
41	Fritz T E, Hefner J E, and Raleigh T M	A Network of Computers Running the UNIX System
42	McGann, L Brian	Application-Level Gateways
43	Hac,Anna	Distributed File Systems - A Survey
44	Weihl,William E.	Data-Dependent Concurrency Control and Recovery
45	Allchin, J. E. and McKendry, M. S.	Synchronization and Recovery of Actions
46	Svobodova, Liba ,ed.	Workshop Summary-Operating Systems in
47	Mohan,C. & Lindsay,B.	Efficient Commit Protocols for the Tree of Processes Model of Distributed Transactions
48	Stankovic, J. et al	Review of Research and Critical Issues in Distributed System Software
49	Bernstein, P. and Goodman, N.	Concurrency in Distributed Data Bases
50	Cheriton,D and Zwaenpoel, W	The Distributed V Kernal and its Performance for Diskless Worksatations
51	Zwaenpoel, W	Implementaion and Performance of Pipes in V System
52	Gaglianello, R and Katseff, H	MEGLOS: An Operating System for a Multiprocessor Environment
53	Powell, M and Miller, B	Process Migration in DEMOS/MP
54	McGann, L	Courier Protocol Access Services with Remote Procedure Call
55	White, J.	Distributed, Replicated Software Locates

NASA/GSFC : Network Operating Systems Bibliography of Relevant Literature

Document #	Source Publication	Date	Key Words/Phrases
26	ACM Computer Communications Review	June 1984	NOS,distributed computer systems
27	Systems & Software	March 1983	ISO OSI,Presentation layer,network,virtual processing system
28	Systems & Software	March 1983	Internetwork,server,distributed database
29	Systems & Software	March 1983	Local network,distributed computing,Apollo Domain
30	Systems & Software	March 1983	Server,Ethernet,virtual disks
31	Systems & Software	March 1983	Electronic mail,centralized/distributed systems,reliability,system recovery
32	Systems & Software	March 1983	ISO Layer 3,broadband LAN,routing
33	Systems & Software	March 1983	ISO Transport layer,ISO L1-4 statistics, datagram,virtual circuit,LAN,long-haul
34	Systems & Software	March 1983	ISO OSI,local network,user programmable NIU
35	Systems & Software	March 1983	Distributed processors,communications-oriented operating system,upper level protocol
36	Systems & Software	March 1983	ISO OSI,networking
37	Micro Communications	March 1985	Novell NetWare, DOS 3.1, LAN O/S, File server
38	ACM Operating Systems Review - Vol 17,#5	June 1983	Distributed O/S,Unix,Ethernet,
39	ACM Operating Systems Review - Vol 15,#3	July 1981	Distributed O/S,Unix,Server
40	Micro Communications	January 1985	Broadband LAN, Remote Operating System Access
41	AT&T Bell Laboratories Technical Journal- Vol 63, No 8	October 1984	UNIX,high-speed LAN,Heterogeneous processor communication
42	Systems & Software	September 1984	LAN's,Application-layer protocol,network file management system
43	ACM Operating Systems Review - Vol 19,#1	January 1985	
44	ACM Operating Systems Review - Vol 19,#1	January 1985	
45	ACM Operating Systems Review - Vol 19,#1	January 1985	
46	ACM Operating Systems Review - Vol 19,#2	April 1985	
47	ACM Operating Systems Review - Vol 19,#2	April 1985	
48	IEEE Distributed Processing Newsletter	March 1985	
49	ACM Computing Surveys	June 1982	
50	ACM Operating Systems Review - Vol 17,#5 9th ACM Symposium on Operating Systems Principles		
51	Proceedings of the 5th International Conference on Distributed Computing Systems	May 1985	
52	Proceedings of the 5th International Conference on Distributed Computing Systems	May 1985	
53	ACM Operating Systems Review - Vol 17,#5 9th ACM Symposium on Operating Systems		
54	Systems and Software	March 1983	
55	Systems and Software	March 1983	

Document #	Author(s)	Title
		Network Resource
56	Pope,A	Encoding CCITT X.409 Presentation Transfer Syntax
57	Foudriat, E.	Embedded Network Computer Research for Future Space Systems
58	Paker, Y	Multiprocessor Systems
59	Andrews, G and Schreider, F	Concepts and Notations for Concurrent Programming
60	Kohler, W	A Survey of Techniques for Synchronization and Recovery in Decentralized Computer Systems
61	Parr, F., J. Auerback and B. Goldstein	Distribute Processing Involving Personal Computers
62	Baker D, J. Wieselthier, and A. Ephremides	Distributed Control Algorithm for Scheduling the Activation of Links in a Self Organizing Mobile Radio Network
63	Desbrisay, G. et al	Airborne Communications Reconstitution Experiment
64	Banino, J. S. et al	Some Fault Tolerant Aspects of the CHORUS Distributed System
65	Miller, B., C. Macrander, and S. Sechrest	A Distributed Programs Monitor for Berkeley UNIX
66	Wittei, L. and R. Curtis	Time Management for Debugging Distributed Systems
67	Fank, A.,L. Wittie, and A. Bernstein	Multicast Communication on Network Computers
68	LeBanc, T. and R. Cook	High Level Broadcast Communication in Local Area Networks
69	Mohan, C. and L. Wittie	Local Reconfiguration of Management Trees in Large Networks

Document #	Source Publication	Date	Key Words/Phrases
56	Computer Communication Review	October 1984	
57	(to be published in IEEE Distributed Processing Technical Committee Newsletter)		
58	Academic Press	1983	
59	ACM Computing Surveys	March 1983	
60	ACM Computing Surveys	June 1981	
61	IEEE Journal on Selected Areas in Communications	May 1983	
62	International Conference on Communications	1982	
63	EASCON 1984		
64	The 5th International Conference on Distributed Computing Systems	1985	
65	The 5th International Conference on Distributed Computing Systems	1985	
66	The 5th International Conference on Distributed Computing Systems	1985	
67	IEEE Software	May 1985	
68	IEEE Software	May 1985	
69	The 5th International Conference on Distributed Computing Systems	1985	

APPENDIX C

-

Bibliography of Relevant Standards

PROTOCOL REFERENCES

LEGEND

n - working memos
dp - draft proposal
dis - draft international standard
is - international standard

GENERAL

ISO International Standard 7498 Open Systems Interconnection Reference Model (also CCITT X.200)

APPLICATION LAYER

ISO n160 Architectural Detail of the Upper Three Layers of OSI

ISO DP 8649/1 OSI Definition of Common Application Service Elements Part I: Introduction

ISO DP 8649/3 OSI Definition of Common Application Service Elements Part III: Commitment Concurrency and Recovery

ISO DP 8650/3 OSI Specification of Protocols for Common Application Service Elements Part III: Commitment Concurrency and Recovery

ISO DP 8650/1 OSI Specification of Protocols for Common Application Service Elements Part I: Introduction

CCITT E.410 International Network Management - General Information.

CCITT E.411 International Network Management - Operational Guidance.

CCITT E.412 International Network Management - Planning.

CCITT E.413 International Network Management - Organization.

ISO n278 Reference Model for Text Preparation and Interchange

ISO DP 8613/4 Text Preparation Text Structures Part IV: Office Document Interchange Formats

ISO DP 8613/3 Text Preparation Text Structures Part III: Document Profile

ISO DP 8613/2 Text Preparation Text Structures Part II: Office Document Architecture

ISO n63 Directory Service for OSI Systems Draft Service Specification

ISO n61 Management Information Service Definition

ISO DP 8832 OSI Specification of the Basic Class Protocol for Job Transfer and Manipulation

ISO DP 8831 OSI Job Transfer and Manipulation Concepts and Services

ISO n175 OSI Virtual Terminal Service -- Basic Class

ISO n223 OSI Virtual Terminal Protocol -- Basic Class

ISO DP 8571/1 OSI File Transfer Access and Management Part I: General Description

ISO DP 8571/2 OSI File Transfer Access and Management Part II: The Virtual Filestore

ISO DP 8571/3 OSI File Transfer Access and Management Part III: The File Service Definition

ISO DP 8571/4 OSI File Transfer Access and Management Part IV: The File Protocol Specification

CCITT X.401, Message Handling Systems: Basic Service Elements and Optional User Facilities.

CCITT X.411, Message Handling Systems: Message Transfer Layer.

CCITT X.420, Message Handling Systems: Interpersonal Messaging User Agent Layer.

CCITT X.409, Message Handling Systems: Presentation Transfer Syntax and Notation.

CCITT X.408, Message Handling Systems: Encoded Information Type Conversion Rules.

CCITT X.430, Message Handling Systems: Access Protocol for Teletex Terminal.

NETWORK LAYER

CCITT X.121 International Numbering Plan for Public Data Networks.

CCITT X.25 Interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks.

ISO DP 8473 Protocol for Providing the Connectionless Network Service.

APPENDIX D

RESEARCH PROJECTS

CNET

N. Lijtmaer and P. Inverardi, IEI-CNR Italy
Distributed system implementation in Ada

CRONUS

Bolt, Beranek, and Newman, Cambridge, MA
Distributed Operating System with heterogeneous local operating system, DOD environment

DEMOS

Michael Powell and Barton Miller, University of California
Process Migration

DISTOS

Dietov Wybranietz, University of Kaiserslautern, W. Germany
Distributed Interprocess Communication, LADY programming environment

ENCHERE

J.-P. Banatre
IRISA (Institut de Recherche pour Information et Systems Aleatoires
High Availability, Non-volatile RAM

IBM SAN JOSE

Bruce Lindsay
Efficiency of Concurrency and Synchronization

IBM SWITZERLAND

Phil Janson, IBM Research
Efficient Inter-Layer (OSI) Communication

MAYFLOWER

Graham Hamilton, University of Cambridge, UK
Remote Procedure Call, Run time binding

RESEARCH PROJECTS

ARCHONS

Douglas Jenson, L. Sha, Carnegie Mellon
Tightly coupled distributed system, high availability, modularity, optimal resource management

ATHENA

Jerry Saltzer, MIT
Network of diverse personal computers, clusters of file and disk servers

BOLT, BERANEK, AND NEWMAN (BBN) - X.409

Arthur Pope
X.409 Data Representation Utility Software

CARNEGIE-MELLON - LU6.2

J. Rosenberg, Carnegie-Mellon and IBM
Program-Program Communication, LU6.2 under UNIX

CARNEGIE-MELLON - IBM Network File System

Alfred Spector
Distributed Verion of ANDREW, move server functions to personal computer, UNIX interface

CHORUS

J.-S. Banino
INRIA (Institute National de Recherche en Informatique et Automatique)
LeChesnay Cedex, France
Interprocess Communication, High Availability

CLOUDS

Martin McKendry
Georgia Tech
Concurrency and Synchronization Algorithms, high availability

MEGLOS

R. Gaglianello and H. Kafseff, AT&T Bell Laboratories
Job Migration in UNIX

MICROS

Larry Wittie, SUNY Stoney Brook
Interprocess Communication, High Availability, Distributed System Debugging

PULSE

Andrew Wellings, University of York, UK
Distributed System Implementation in Ada, file system, interprocess communication

SWIFT

Larry Swift, MIT
Interprocess Communication, Upcalls

DISTRIBUTED V SYSTEM

David Cheriton, Stanford University
Distributed System Performance, Efficiency Distributed Interprocess Communication, Process Groups

NASA Langley

Edwin Foudriat
High Availability, Multipathing

DISTRIX

David Christie, Convergent Technologies
UNIX System V, message based kernel, single user workstations network

LINCS

John Fletcher, Lawrence Livermore National Laboratory
High throughput networks

CAMBRIDGE DISTRIBUTED SYSTEM/PROJECT UNIVERSE

Roger Needham, University of Cambridge, UK

ANDREW

M. Satya
Very large distributed file system (to 5000 nodes), UNIX file systems
Single, system wide shared name space

ARGUS

William E. Weihl, MIT
Concurrency and Synchronization