NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

TCP-IP BASED ON CLIENT SERVER COMPUTING

Graduation Project
Com400

Supervisor : Prof Dr. Doğan Ibrahim

Student : Önder Karlıoğlu

Nicosia 2004

TABLE OF CONTENTS

INTRODUCTION

'Client/Server' is one of the computer industries newest and hottest buzz words. There is no generic definition of client/server as it is used to depict a vast number of mature, developing, and anticipated technologies. However the general idea is that clients and servers are separate logical entities that work together over a network to accomplish a task (Edwards, Harkey, and Orfali, 1996).

The parts of a client/server system are the client, server, and the network. The client can be a personnel computer (PC) or the soon to be released network computer (NC.) The server can be anything from a mainframe to a PC but is usually a platform suited for a specific application or task. The network could be as simple as two computers connected by modem or as complicated as several thousand computers linked by a combination of fiber optics, coaxial cab and wireless communications. The client and server might be in the same room (possibly even the same physical machine) or half a world apart. The difference between client/server computing and the mainframe computing that it often supersedes is that the client is not a dumb terminal. The client has its own operating system and can manage inputs (keyboard, mouse, etc..) and outputs (display, local printer, sound, etc..) without the server. The server's role is to wait passively for a client to request a service. This distribution of processing allows the client to offer a more user friendly environment (graphical user interface, local applications, mouse, etc..) than a dumb terminal and allows the servers to be less complex and expensive than current mainframe systems. Overall client/server computing leads to a flexible and dynamic computing environment.

This brief introduction might give the impression that client/server computing has many advantages over the mainframe computing that it often replaces. The hype surrounding client/server technology and the computing industry in general tend to push consumers in the direction of client/server technology. However there are some very real issues that a chief information officer (CIO) must contemplate when considering a change to client/server computing. This essay will examine the cost, sustainability, user productivity, reliability, and security of client/server computing as compared to mainframe computing.

PREFACE

The client/server computing model defines the way *successful* organizations will use technology during the next decade. It is the culmination of the trend toward downsizing applications :from the minicomputer and mainframe to the desktop. Enabling technologies, such as object-oriented development and graphical user interfaces (Gills), will liberate the users and owners of information to use technology personally and directly. Users will no longer need continual assistance from professional information systems (IS) personnel to create and store their business data.

The big losers in this change will be traditional vendors and integrators of minicomputer-based solutions. Forrester Research Inc., a reputable computer industry market research firm, routinely surveys the U.S. Fortune 1000 companies. Forrester projects that by 1993 the client/server market will account for $29 billion in sales. The pervasiveness of this technology throughout organizations dictates that all management levels understand the concepts and implications of client/server computing. Information systems (IS) professionals must understand these concepts and implications, as well as detailed architectural issues involved, in order to be in a position to offer liberating client/server solutions to their users. IS professionals who do not understand these concerns will be relegated forever to a maintenance role on existing systems.

---1 ---

Client/Server Computing

1.1 Client/Server Computing

The single-system image is best implemented through the client/server model. Our experience confirms that client/server computing can provide the enterprise to the desktop. Because the desktop computer is the user's view into the enterprise, there is no better way to guarantee a single image than to start at the desktop.

Unfortunately, it often seems as if the number of definitions of client/server computing depends on how many organizations you survey, whether they're hardware and software vendors, integrators, or IS groups. Each has a vested interest in a definition that makes its particular product or service an indispensable component.

Throughout this book, the following definitions will be used consistently:

Client: A client is a single-user workstation that provides presentation services and the appropriate computing, connectivity, and database services and interfaces relevant to the business need.

Server: A server is one or more multiuser processors with shared memory providing computing, connectivity, and database services and interfaces relevant to the business need.

Client/server computing is an environment that satisfies the business need by appropriately allocating the application processing between the client and the server processors. The client requests services from the server; the server processes the request and returns the result to the client. The communications mechanism is a message passing *interprocess communication* (IPC) that enables (but does not require) distributed placement of the client and server processes. Client/server is a software model of computing, not a hardware definition.

This definition makes client/server a rather generic model and fits what is known in the industry as "cooperative processing" or "peer-to-peer."

Because the client/server environment is typically heterogeneous, the hardware platform and operating system of the client and server are not usually the same. In such cases, the communications mechanism may be further extended through a well-defined set of standard application program interfaces (APis) and remote procedure calls (RPCs).

The modern diagram representing the client/server model was probably first popularized by Sybase.

Effective client/server computing will be fundamentally platform-independent. The user of an application wants the business functionality it provides; the computing platform provides access to this business functionality. There is no benefit, yet considerable risk, in exposing this platform to its user.

Changes in platform and underlying technology should be transparent to the user. Training costs, business processing delays and errors, staff frustration, and staff turnover result from the confusion generated by changes in environments where the user is sensitive to the technology platform.

It is easily demonstrated that systems built with transparency to the technology, for all users, offer the highest probability of solid ongoing return for the technology investment. It is equally demonstrable that if developers become aware of the target platform, development will be bound to that platform. Developers will use special features, tricks, and syntax found only in the specific development platform.

Tools, which isolate developers from the specifics of any single platform, assist developers in writing transparent, portable applications. These tools must be available for each of the three essential components in any application: data access, processing, and interfaces. Data access includes the graphical user interface (GUI) and stored data access. Processing includes the business logic. Interfaces link services with other applications. This simple model, reflected in Figure 1.5, should be kept in mind when following the evolution to client/server computing.

The use of technology layers provides this application development isolation. These layers isolate the characteristics of the technology at each level from the layer above and below. This layering is fundamental to the development of applications in the client/server model. The rapid rate of change in these technologies and the lack of experience with the "best" solutions implies that we must isolate specific technologies from each other. This book will continue to emphasize and expand on the concept of a *systems development environment* (SDE) as a way to achieve this isolation.

Developer tools are by far the most visible. Most developers need to know only the syntax of these tools to express the business problem in a format acceptable to the technology platform. With the increasing involvement of noncomputer professionals, as technology users and application assemblers, technology isolation is even more important. Very few-perhaps none-of an organization's application development staff needs to be aware of the hardware, system software, specific database engines, specific communications products, or specific presentation services products. These are invoked through the APis message passing, and RPCs generated by tools or by a few technical specialists.

As organizations increase the use of personal productivity tools, workstations become widely installed. The need to protect desktop real estate requires that host terminal capabilities be provided by the single workstation. It soon becomes evident that the power of the workstation is not being tapped and application processing migrates to the desktop. Once most users are connected from their workstation desktop to the applications and data at the host mainframe or minicomputer, there is significant cost benefit in offloading processing to these powerful workstations. The first applications tend to be data capture and edit. These simplify-but still use-the transaction expected by an already existing host application. If the workstation is to become truly integrated with the application, reengineering of the business process will be necessary. Accounting functions and many customer service applications are easily offloaded in this manner. Thus, workgroup and departmental processing is done at the LAN level, with host involvement for enterprise-wide data and enforcement of interdepartmental business rules.

In this "dumb" terminal (IBM uses the euphemism *nonprogrammable* to describe its 327x devices) emulation environment, all application logic resides in the minicomputer, mainframe, or workstation. Clearly a $5000 or less desktop workstation is capable of much more than the character display provided by a $500 terminal. In the client/server model, the low-cost processing power of the workstation will replace host processing, and the application logic will be divided appropriately among the platforms. As previously noted, this distribution of function and data is transparent to the user and application developer.

1.2 Mainframe-Centric Client/Server Computing

The mainframe-centric model uses the presentation capabilities of the workstation to front-end existing applications. The character mode interface is remapped by products such as Easel and Mozart. The same data is displayed or entered through the use of pull-down lists, scrollable fields, check boxes, and buttons; the user interface is easy to use, and information is presented more clearly. In this mainframe-centric model, mainframe applications continue to run unmodified, because the existing terminal data stream is processed by the workstation-based communications API.

The availability of products such as UniKix and IBM's CICS OS/2 and 6000 can enable the entire mainframe processing application to be moved unmodified to the workstation. This protects the investment in existing applications while improving performance and reducing costs.

Character mode applications, usually driven from a block mode screen, attempt to display as much data as possible in order to reduce the number of transmissions required to complete a function. Dumb terminals impose limitations on the user interface including fixed length fields, fixed length lists, crowded screens, single or limited character fonts, limited or no graphics icons, and limited windowing for multiple application display. In addition, the fixed layout of the screen makes it difficult support the display of conditionally derived information.

In contrast, the workstation GUI provides facilities to build the screen dynamically. This enables screens to be built with a variable format based conditionally on the data values of specific fields. Variable length fields can be scrollable, and lists of fields can have a scrollable number of rows. This enables a much larger virtual screen to be used with no additional data communicated between the client workstation and server.

Windowing can be used to pull up additional information such as help text, valid value lists, and error messages without losing the original screen contents. The more robust GUI facilities of the workstation enable the user to navigate easily around the screen.

Additional information can be encapsulated by varying the display's colors, fonts, graphics icons, scrollable lists, pull-down lists, and option boxes. Option lists can be provided to enable users to quickly select input values. Help can be provided, based on the context and the cursor location, using the same pull-down list facilities.

Although it is a limited use of client/server computing capability, a GUI front end to an existing application is frequently the first client/server-like application implemented by organizations familiar with the host mainframe and dumb-terminal approach. The GUI preserves the existing investment while providing the benefits of ease of use associated with a GUI. It is possible to provide dramatic and functionally rich changes to the user interface without host application change.

The next logical step is the provision of some edit and processing logic executing at the desktop workstation. This additional logic can be added without requiring changes in the host application and may reduce the host transaction rate by sending up only valid transactions. With minimal changes to the host application, network traffic can be reduced and performance can be improved by using the workstation's processing power to encode the datastream into a compressed form.

A more interactive user interface can be provided with built-in, context-sensitive help, and extensive prompting and user interfaces that are sensitive to the users' level of expertise. These options can be added through the use of workstation processing power. These capabilities enable users to operate an existing system with less intensive training and may even provide the opportunity for public access to the applications.

Electronic data interchange (EDI) is an example of this front-end processing. EDI enables organizations to communicate electronically with their suppliers or customers. Frequently, these systems provide the workstation front end to deal with the EDI link but continue to work with the existing back-end host system applications. Messages are reformatted and responses are handled by the EDI client, but application processing is done by the existing application server. Productivity may be enhanced significantly by capturing information at the source and making it available to all authorized users. Typically, if users employ a multipart form for data capture, the form data is entered into multiple systems. Capturing this information once to a server in a client/server application, and reusing the data for several client applications can reduce errors, lower data entry costs, and speed up the availability of this information.

The data is available to authorized users as soon as it is captured. There is no delay while the forms are passed around the organization. This is usually a better technique than *forms imaging technology* in which the forms are created and distributed internally in an organization. The use of workflow-management technology and techniques, in conjunction with imaging technology, is an effective way of handling this process when forms are filled out by a person who is physically remote from the organization.

Intelligent Character Recognition (ICR) technology can be an extremely effective way to automate the capture of data from a form, without the need to key. Current experience with this technique shows accuracy rates greater than 99.5 percent for typed forms and greater than 98.5 percent for handwritten forms.

1.3 Downsizing and Client/Server Computing

Rightsizing and downsizing are strategies used with the client/server model to take advantage of the lower cost of workstation technology. Rightsizing and upsizing may involve the addition of more diverse or more powerful computing resources to an enterprise computing environment. The benefits of rightsizing are reduction in cost and/or increased functionality, performance, and flexibility in the applications of the enterprise. Significant cost savings usually are obtained from a resulting reduction in employee, hardware, software, and maintenance expenses. Additional savings typically accrue from the improved effectiveness of the user community using client/server technology.

Downsizing is frequently implemented in concert with a flattening of the organizational hierarchy.

Eliminating middle layers of management implies empowerment to the first level of management with the decision-making authority for the whole job. Information provided at the desktop by networked PCs and workstations integrated with existing host (such as mainframe and minicomputer) applications is necessary to facilitate this empowerment. These desktop-host integrated systems house the information required to make decisions quickly. To be effective, the desktop workstation must provide access to this information as part of the normal business practice. Architects and developers must work closely with business decision makers to ensure that new applications and systems are designed to be integrated with effective business processes. Much of the cause of poor return on technology investment is attributable to a lack of understanding by the designers of the day-to-day business impact of their solutions.

Downsizing information systems is more than an attempt to use cheaper workstation technologies to replace existing mainframes and minicomputers in use. Although some benefit is obtained by this approach, greater benefit is obtained by reengineering the business processes to really use the capabilities of the desktop environment. Systems solutions are effective only when they are seen by the actual user to add value to the business process.

Client/server technology implemented on low-cost standard hardware will drive downsizing. Client/server computing makes the desktop the users' enterprise. As we move from the machine-centered era of computing into the workgroup era, the desktop workstation is empowering the business user to regain ownership of his or her information resource. Client/server computing combines the best of the old with the new+: the reliable multiuser access to shared data and resources with the intuitive, powerful desktop workstation.

Object-oriented development concepts are embodied in the use of an SDE created for an organization from an architecturally selected set of tools. The SDE provides more effective development and maintenance than companies have experienced with traditional host-based approaches.

Client/server computing is open computing. Mix and match is the rule. Development tools and development environments must be created with both openness and standards in mind.

Mainframe applications rarely can be downsized-without modifications-to a workstation environment. Modifications can be minor, wherein tools are used to *port* (or rehost) existing mainframe source code-or major, wherein the applications are *rewritten* using completely new tools. In porting, native COBOL compilers, functional file systems, and emulators for DB2, IMS DB/DC, and CICS are available for workstations. In rewriting, there is a broad array of tools ranging from PowerBuilder, Visual Basic, and Access, to larger scale tools such as Forte and Dynasty.

1.4 Preserving Your Mainframe Applications Investment Through Porting

Although the percentage of client/server applications development is rapidly moving away from a mainframe-centric model, it is possible to downsize and still preserve a larger amount of the investment in application code. For example, the Micro Focus COBOL/2 Workbench by Micro Focus Company Inc., and XDB Systems Inc., bundles products from Innovative Solutions Inc., Stingray Software Company Inc., and XDB Systems Inc., to provide the capability to develop systems on a PC LAN for production execution on an IBM mainframe. These products, in conjunction with the ProxMVS product from Proximity Software, enable extensive unit and integration testing to be done on a PC LAN before moving the system to the mainframe for final system and performance testing. Used within a properly structured development environment, these products can dramatically reduce mainframe development costs.

Micro Focus COBOL/2 supports GUI development targeted for implementation with OS/2 Presentation Manager and Microsoft Windows 3.x. Another Micro Focus product, the Dialog System, provides support for GUI and character mode applications that are independent of the underlying COBOL applications.

Micro Focus has added an Object Oriented (OO) option to its workbench to facilitate the creation of reusable components. The OO option supports integration with applications developed under Smalltalk/V PM.

IBM's CICS for OS/2, OS400, RS6000, and HP/UX products enable developers to directly port applications using standard CICS call interfaces from the mainframe to the workstation. These applications can then run under OS/2, AIX, OS400, HP/UX, or MVSNSE without modification. This promises to enable developers to create applications for execution in the CICS MVS environment and later to port them to these other environments without modification. Conversely, applications can be designed and built for such environments and subsequently ported to MVS (if this is.a logical move). Organizations envisioning such a migration should ensure that their SDE incorporates standards that are consistent for all of these platforms.

To help ensure success in using these products, the use of a COBOL code generator, such as Computer Associates' (previously Pansophic) Telon PWS, provides the additional advantages of a higher level of syntax for systems development. Telon provides particularly powerful facilities that support the object-oriented development concepts necessary to create a structured development environment and to support code and function reuse. The generated COBOL is input to the Micro Focus Workbench toolkit to support prototyping and rapid application development. Telon applications be generated to execute in the OS/2, UNIX AIX, OS400, IMS DB/DC, CICS DLI, DB2, IDMS, and Datacom DB environments. This combination-used in conjunction with a structured development environment that includes appropriate standards=-provides the capability to build single-system image applications today. In an environment that requires preservation of existing host-based applications, this product suite is among the most complete for client/server computing.

These products, combined with the cheap processing power available on the workstation, make the workstation LAN an ideal development and maintenance environment for existing host processors. When an organization views mainframe or minicomputer resources as real dollars, developers can usually justify offloading the development in only three to six months. Developers can be effective only when a proper systems development environment is put in place and provided with a suite of tools offering the host capabilities plus enhanced connectivity. Workstation operating systems are still more primitive than the existing host server MVS, VMS, or UNIX operating systems. Therefore, appropriate standards and procedures must be put in place to coordinate shared development. The workstation environment will change. Only projects built with common standards and procedures will be resilient enough to remain viable in the new environment.

The largest savings come from new projects that can establish appropriate standards at the start and do all development using the workstation LAN environment. It is possible to retrofit standards to an existing environment and establish a workstation with a LAN-based maintenance environment. The benefits are less because retrofitting the standards creates some costs. However, these costs are justified when the application is scheduled to undergo significant maintenance or if the application is very critical and there is a desire to reduce the error rate created by changes. The discipline associated with the movement toward client/server-based development, and the transfer of code between the host and client/server will almost certainly result in better testing and fewer errors. The testing facilities and usability of the workstation will make the developer and tester more effective and therefore more accurate.

Business processes use database, communications, and application services. In an ideal world, we pick the best servers available to provide these services, thereby enabling our organizations to enjoy the maximum benefit that current technology provides. Real-world developers make compromises around the existing technology, existing application products, training investments, product support, and a myriad other factors.

Key to the success of full client/server applications is selecting an appropriate application and technical architecture for the organization. Once the technical architecture is defined, the tools are known. The final step is to implement an SDE to define the standards needed to use the tools effectively. This SDE is the collection of hardware, software, standards, standard procedures, interfaces, and training built up to support the organization's particular needs.

1.5 The Real World of Client/Server Development Tools

Many construction projects fail because their developers assume that a person with a toolbox full of carpenter's tools is a capable builder. To be a successful builder, a person must be trained to build according to standards. The creation of standards to define interfaces to the sewer, water, electrical utilities, road, school, and community systems is essential for successful, cost-effective building. We do not expect a carpenter to de such interfaces individually for every building. Rather, pragmatism discourages imagination in this regard. By reusing the models previously built to accomplish integration, we all benefit from cost and risk reduction.

Computer systems development using an SDE takes advantage of these same concepts: Let's build on what we've learned. Let's reuse as much as possible to save development costs, reduce risk, and provide the users with a common "look and feel."

Selecting a good set of tools affords an opportunity to be successful. Without the implementation of a comprehensive SDE, developers will not achieve such success.

The introduction of a whole new generation of Object Technology based tools for client/server development demands that proper standards be put in place to support Shared development, reusable code, interfaces to existing systems, security, error handling, and an organizational standard "look and feel." As with any new technology, there will be changes. Developers can build application systems closely tied to today's technology or use an SDE and develop applications that can evolve along with the technology platform.

---2---

Advantages of Client/Server Computing

2.1 Executive Summary

Organizations want to take advantage of the low- cost and user-friendly environment that existing desktop workstations provide. There is also a strong need and desire to capitalize on existing investment at the desktop and in the portfolio of business applications currently running in the host. Thus, corporate networks are typically put in place to connect user workstations to the host. Immediate benefits are possible by integrating these three technologies: workstations, connectivity, and hosts. Retraining and redevelopment costs are avoided by using the existing applications from an integrated desktop.

Client/server computing provides the capability to use the most cost-effective user interface, data storage, connectivity, and application services. Frequently, client/server products are deployed within the present organization but are not used effectively. The client/server model provides the technological means to use previous investments in concert with current technology options. There has been a dramatic decline in the cost of the technology components of client/server computing. Organizations see opportunities to use technology to provide business solutions. Service and quality competition in the marketplace further increase the need to take advantage of the benefits available from applications built on the client/server model.

Client/server computing in its best implementations moves the data-capture and information-processing functions directly to the knowledgeable worker-that is, the worker with the ability to respond to errors in the data, and the worker with the ability to use the information made available. Systems used in the front office, directly involved in the process of doing the business, are forced to show value. If they don't, they are discarded under the cost pressures of doing business. Systems that operate in the back room after the business process is complete are frequently designed and implemented to satisfy an administrative need, without regard to their impact on business operations. Client/server applications integrate the front and back office processes because data capture and usage become an integral part of the business rather than an after-the-fact administrative process. In this mode of operation, the processes are continuously evaluated for effectiveness. Client/server computing provides the technology platform to support the vital business practice of continuous improvement.

2.2 The Advantages of Client/Server Computing

The client/server computing model provides the means to integrate personal productivity applications for an individual employee or manager with specific business data processing needs to satisfy total information processing requirements for the entire enterprise.

2.2.1 Enhanced Data Sharing

Data that is collected as part of the normal business process and maintained on a server is immediately available to all authorized users. The use of Structured Query Language (SQL) to define and manipulate the data provides support for open access from all client processors and software. SQL grants all authorized users access to the information through a view that is consistent with their business need. Transparent network services ensure that the same data is available with the same currency to all designated users.

2.2.2 Integrated Services

In the client/server model, all information that the client (user) is entitled to use is available at the desktop. There is no need to change into terminal mode or log into another processor to access information. All authorized information and processes are directly available from the desktop interface. The desktop tools-e-mail, spreadsheet, presentation graphics, and word processing-are available and can be used to deal with information provided by application and database servers resident on the network. Desktop users can use their desktop tools in conjunction with information made available from the corporate systems to produce new and useful information.

A word-processed document that includes input from a drawing package, a spreadsheet, and a custom-developed application can be created. The facilities of Microsoft's Dynamic Data Exchange (DDE) enable graphics and spreadsheet data to be cut and pasted into the word-processed document along with the window of information extracted from a corporate database. The result is displayed by the custom application.

Creation of the customized document is done using only desktop tools and the mouse to select and drag information from either source into the document. The electronic scissors and glue provide powerful extensions to existing applications and take advantage of the capability of the existing desktop processor. The entire new development can be done by individuals who are familiar only with personal productivity desktop tools. Manipulating the spreadsheet object, the graphics object, the application screen object, and the document object using the desktop cut and paste tools provides a powerful new tool to the end user.

Developers use these same object manipulation capabilities under program control to create new applications in a fraction of the time consumed by traditional programming methods. Object-oriented development techniques are dramatically increasing the power available to nonprogrammers and user professionals to build and enhance applications.

Another excellent and easily visualized example of the integration possible in the client/server model is implemented in the retail automobile service station. The service station automation (SSA) project integrates the services of gasoline flow measurement, gas pumps billing, credit card validation, cash registers management, point-of-sale, inventory control, attendance recording, electronic price signs, tank monitors, accounting, marketing, truck dispatch, and a myriad of other business functions. These business functions are all provided within the computer-hostile environment of the familiar service station with the same type of workstations used to create this book. The system uses all of the familiar client/server components, including local and wide-area network services. Most of the system users are transitory employees with minimal training in computer technology. An additional challenge is the need for real-time processing of the flow of gasoline as it moves through the pump. If the processor does not detect and measure the flow of gasoline, the customer is not billed. The service station automation system is a classic example of the capabilities of an integrated client/server application implemented and working today.

2.2.3 Sharmg Resources Among Diverse Platforms

The client/server computing model provides opportunities to achieve true open system computing. Applications may be created and implemented without regard to the hardware platforms or the technical characteristics of the software. Thus, users may obtain client services and transparent access to the services provided by database, communications, and applications servers. Operating systems software and platform hardware are independent of the application and masked by the development tools used to build the application.

In this approach, business applications are developed to deal with business processes invoked by the existence of a user-created "event." An event such as the push of a button, selection of a list element, entry in a dialog box, scan of a bar code, or flow of gasoline occurs without the application logic being sensitive to the physical platforms.

Client/server applications operate in one of two ways. They can function as the front end to an existing application-the more limited mainframe-centric model discussed in Chapter 1-or they can provide data entry, storage, and reporting by using a distributed set of clients and servers. In either case, the use-or even the existence-of a mainframe host is totally masked from the workstation developer by the use of standard interfaces such as SQL.

2.2.4 Data Interchangeability and Interoperability

SQL is an industry-standard data definition and access language. This standard definition has enabled many vendors to develop production-class database engines to manage data as SQL tables. Almost all the development tools used for client/server development expect to reference a back-end database server accessed through SQL. Network services provide transparent connectivity between the client and local or remote servers. With some database products, such as Ingres Star, a user or application can define a consolidated view of data that is actually distributed between heterogeneous, multiple platforms.

Systems developers are finally reaching the point at which this heterogeneity will be a feature of all production-class database engine products. Most systems that have been implemented to date use a single target platform for data maintenance. The ability to do high-volume updates at multiple locations and maintain database integrity across all types of errors is just becoming available with production-level quality performance and recovery. Systems developed today that use SQL are inherently transparent to data storage location and the technology of the data storage platform. The SQL syntax does not specify a location or platform. This transparency enables tables to be moved to other platforms and locations without affecting the application code. This feature is especially valuable when adopting proven, new technology or if it makes business sense to move data closer to its owner.

Database services can be provided in response to an SQL request-without regard to the underlying engine. This engine can be provided by vendors such as ASK/Ingres, Oracle, Sybase, or IBM running on Windows NT, OS/2, UNIX, or MVS platform. The system development environment (SDE) and tools must implement the interfaces to the vendor database and operating system products. The developer does not need to know which engine or operating system is running. If the SDE does not remove the developer from direct access to the database server platform, the enthusiasm to be efficient will prevent developers from avoiding the use of "features" available only from a specific vendor. The transparency of platform is essential if the application is to remain portable. Application portability is essential when taking advantage of innovation in technology and cost competitiveness, and in providing protection from the danger of vendor failure.

Database products, such as Sybase used with the Database Gateway product from Micro DecisionWare, provide direct, production-quality, and transparent connectivity between the client and servers. These products may be implemented using DB2, IMS/DB, or VSAM through CICS into DB2, and Sybase running under VMS, Windows NT, OS/2, DOS, and MacOS. Bob Epstein, executive vice president of Sybase, Inc., views Sybase's open server approach to distributed data as incorporating characteristics of the semantic heterogeneity solution. 1 In this solution, the code at the remote server can be used to deal with different database management systems (DBMSs), data models, or processes. The remote procedure call (RPC) mechanism used by Sybase can be interpreted as a message that invokes the appropriate method or procedure on the open server. True, somebody has to write the code that masks the differences. However, certain parts-such as accessing a foreign DBMS (like Sybase SQL Server to IBM DB2)-can be standardized.

ASK's Ingres Star product provides dynamic SQL to support a distributed database between UNIX and MVS. Thus, Ingres Windows 4GL running under DOS or UNIX as a client can request a data view that involves data on the UNIX Ingres and MVS DB2 platform. Ingres is committed to providing static SQL and IMS support in the near future. Ingres' Intelligent Database engine will optimize the query so that SQL requests to distributed databases are handled in a manner that minimizes the number of rows moved from the remote server. This optimization is particularly crucial when dynamic requests are made to distributed databases. With the announcement of the Distributed Relational Database Architecture (DRDA), IBM has recognized the need for open access from other products to DB2. This product provides the app-lication program interfaces (APIs) necessary for other vendors to generate static SQL requests to the DB2 engine running under MVS. Norris van den Berg, manager of Strategy for Programming Systems at IBM's Santa Teresa Laboratory in San Jose, California, points out that IBM's Systems Application Architecture (SAA) DBMSs are different. Even within IBM, they must deal with the issues of data interchange and interoperability in a heterogeneous environment.2 More importantly, IBM is encouraging third-party DBMS vendors to comply with its DRDA This is a set of specifications that will enable all DBMSs *to* interoperate.

The client/server model provides the capability to make ad hoc requests for information. As a result, optimization of dynamic SQL and support for distributed databases are crucial for the success of the second generation of a client/server application. The first generation implements the operational aspects of the business process. The second generation is the introduction of ad hoc requests generated by the knowledgeable user looking to gain additional insight from the information available.

2.2.5 Masked Physical Data Access

When SQL is used for data access, users can access information from databases anywhere in the network. From the local PC, local server, or wide area network (WAN) server, data access is supported with the developer and user using the same data request. The only noticeable difference may be performance degradation if the network bandwidth is inadequate. Data may be accessed from dynamic random-access memory (D-RAM), from magnetic disk, or from optical disk, with the same SQL statements. Logical tables can be accessed-without any knowledge of the ordering of columns or awareness of extraneous columns-by selecting a subset of the columns in a table. Several tables may be joined into a view that creates a new logical table for application program manipulation, without regard to its physical storage format.

The use of new data types, such as binary large objects (BLOBs), enables other types of information such as images, video, and audio to be stored and accessed using the same SQL statements for data access. RPCs frequently include data conversion facilities to translate the stored data of one processor into an acceptable format for another.

2.2.6 Location Independence of Data and Processing

We are moving from the machine-centered computing era of the 1970s and 1980s to a new era in which PC-familiar users demand systems that are user-centered. Previously, a user logged into a mainframe, mini-, or microapplication. The syntax of access was unique in each platform. Function keys, error messages, navigation methods, security, performance, and editing were all very visible. Today's users expect a standard "look and feel." Users log into an application from the desktop with no concern for the locatiou or technology of the processors involved.

Developers today are provided with considerable independence. Data is accessed through SQL without regard to the hardware, operating system, or location providing the data. Consistent network access methods envelop the application and SQL requests within an RPC. The network may be based in Open Systems Interconnect (OSI), Transmission Control Protocol/Internet Protocol (TCP/IP), or Systems Network Architecture (SNA), but no changes are required in the business logic coding. The developer of business logic deals with a standard process logic syntax without considering the physical platform. Development languages such as COBOL, C, and Natural, and development tools such as Telon, Ingres 4GL, PowerBuilder, CSP, as well as some evolving CASE tools such as Bachman, Oracle CASE, and Texas Instruments' IEF all execute on multiple platforms and generate applications for execution on multiple platforms.

The application developer deals with the development language and uses a version of SDE customized for the organization to provide standard services. The specific platform characteristics are transparent and subject to change without affecting the application syntax.

2.2.7 Centralized Management

As processing steers away from the central data center to the remote office and plant, workstation server, and local area network (LAN) reliability must approach that provided today by the centrally located mini- and mainframe computers. The most effective way to ensure this is through the provision of monitoring and support from these same central locations. A combination of technologies that can "see" the operation of hardware and software on the LAN-monitored by experienced support personnel-provides the best opportunity to achieve the level of reliability required.

The first step in effectively providing remote LAN management is to establish standards for hardware, software, networking, installation, development, and naming. These standards, used in concert with products such as IBM's Systemview, Hewlett-Packard's Openview, Elegant's ESRA, Digital's EMA, and AT&T's UNMA products, provide the remote view of the LAN. Other tools, such as PC Connect for remote connect, PCAssure from Centel for security, products for hardware and software inventory, and local monitoring tools such as Network General's Sniffer, are necessary for completing the management process.

2.3 Technology Revolution

The changes in computer technology that have taken place during the past five years are significantly greater than those of the preceding 35 years of computer history. There is no doubt that we will continue to experience an even greater rate of change during the coming five-year period.

2.3.1 Future Technologies

Consulting a crystal ball, projecting the future, and making decisions based on the projections is a common failure of the computer industry, Predicting the future is a risky business. Industry leaders, technicians, and investors have been equally unsuccessful on occasion. It is important, however, to achieve an educated view of where technology is headed during the life of a new system. The architecture on which a new system is built must be capable of supporting all users throughout its life. Large organizations traditionally have assumed that their applications will provide useful service for 5 to 10 years. Many systems are built with a view of only what is available and provable today, and they are ready to fall apart like a deck of cards when the operating environment changes and the architecture cannot adapt to the new realities. Properly architected systems consider not only the reality of today but also an assessment of the likely reality five years after the date of implementation.

Despite predictions that the scope of change in computer technology in the next five years will exceed that seen in the entire computer era (1950 through 1994), a view of history still provides the only mirror we have into the future.

2.3.2 Computing Power Compared

A 1990 survey of U.S. Fortune 1000 companies, completed by a well-known computer industry research firm, found that on an MIPS (millions of instructions per second) basis, more than 90 percent of the processing power available to organizations exists at the desktop. This cheap computing power is typically underused today. It is a sunk cost available to be used as clients in the implementation of client/server applications.

2.3.3 Input/Output (I/0) Compared

Most applications require information that is manipulated also to be read and saved. In the next example, added to the CPU processing is the requirement to perform 1000 physical data read or write operations per second.

The same portion of the mainframe configuration required to provide one MIPS execution capability can simultaneously handle this I/O requirement. The workstation configuration required to simultaneously handle these two tasks in 1989 cost at least twice that of the mainframe configuration. In addition, the configuration involved multiple processors without shared memory access. In order to preserve data integrity, the I/O must be read only. The dramatic reduction in workstation cost projected in 1995 is predicated on the use of symmetric multiprocessors to provide CPUs with shared memory and on the use of coprocessors providing the cached controllers necessary to support parallel I/0. (Parallel I/O enables multiple I/O requests to several devices to be serviced concurrently with host CPU processing.) However, the costs are still projected to be 75 percent greater than costs on the mainframe for this high rate of I/O.

The difference in price and functionality is primarily explained by the fact that the IBM 3090-600 is an example of a massively parallel processor optimized to do I/0. Every channel, DASD controller, tape controller, and console contains other processors. The processing capacity of these other processors is three to eight times the processing capacity of the main processor. These processors have direct memory access (DMA) to the shared memory of the configuration, with minimal impact on the processing capacity of the main processor. These processors enable I/O operations to proceed in parallel with little or no main processor involvement.

For the immediate future, forecasts show little slackening in demand for large host processors to provide enterprise database engine services for large companies, especially Fortune 500 firms. Ad hoc processing demands generated by the availability of workplace requestors will further increase the I/O demand. The R.ISC and Intel processors, as configured today and envisioned over the next five years, continue to use the main processor to perform much of the processing involved in I/O functions. This is an economical strategy for most client applications and many server applications where the I/O demands do not approach those found in large host mainframe configurations. Distributed database technology reduces the demands for I/O against a single database configuration and distributes the I/O with the data to the remote server processors. Despite the dramatic increase in CPU power, there hasn't been a corresponding increase in the capability to do "real" I/0. Some mechanical limitations are not solved by increased CPU power. In fact, the extra CPU merely enables I/O requests to be generated more rapidly.

Only through the effective use of real storage (D-RAM) can we hope to use the available CPU power. Data can be accessed from D-RAM without the need to do physical I/O except to log the update. Database technology uses a sequential log to record changes. These sequential writes can be buffered and done very rapidly. The random updates to the database are done when the system has nothing better to do or when the shared D-RAM containing the updated data is required for other data. The log is used to recover the database after any failure that terminates the application.

Another complication in the I/O factor is the steadily decreasing cost of permanent data storage devices. As the cost of traditional data storage devices-disk and tape-decreases, new technologies with massively greater capacity have evolved. Optical storage devices provide greater storage for less cost but with a somewhat slower rate of access than magnetic disk technologies. Most industry experience demonstrates that the amount of data an organization wants to store depends on the cost of storage, not on any finite limit to the amount of data available. If the cost of storage is halved, twice as much data will be available to store for the same budget. This additional data may come from longer histories, external sources, or totally new forms of data, such as image, audio, video, and graphics. New applications may be justified by the reduction in cost of data stores.

Workstation technologies can deal with personal data, data extracted from central systems for analysis by the end user, data from integrated external sources for comparison, and integrated new types of data such as voice annotation to documents. All these data forms provide additional uses for lower-cost, permanent data storage. Decision-support systems can use workstation technologies and massive amounts of additional data to provide useful, market-driven recommendations.

Relational database technologies also can limit the amount of real I/O required to respond to information requests. The use of descriptor indexes that contain data values extracted from columns of the database tables enables search criteria to be evaluated by accessing only the indexes. Access to the physical database itself is required only when the index search results in the identification of rows from the relational table that satisfy the search criteria. Large relational tables, which are accessed through complex searches, can demonstrate dramatically different performance and cost of access depending on the effectiveness of the database search engine. Products such as DB2 and Ingres, which do extensive query optimization, often demonstrate significantly better performance than other products in complex searches. Products that were developed to deal with a small memory model often exhibit dramatic CPU overhead when the size of resident indexes gets very large. DB2 achieves linear improvement in performance as indexes are allocated more D-RAM. Oracle, on the other hand, does not perform well in the IBM System 370 MVS implementation because of its overhead in managing very large main storage buffer pools.

2.3.4 Main Storage

Arguably, the most dramatic technological revolution affecting the computer industry today is caused by the increase in the amount of main storage (D-RAM) available to an application. D-RAM is used for the execution of programs and the temporary storage of permanent data.

Computer users have entered the era of very large and inexpensive D-RAM. Figure 2.10 represents the manner in which this technology has evolved and continues to evolve. Every three years, a new generation of D-RAM technology is released. Each new generation is released with four times the capacity of the previous generation for the same chip price. At the point of introduction and at any given time during its life cycle, the cost of these chips is reduced to a price equal to the price of chips from the previous generation. As the capacity of individual D-RAM chips has increased, the quantity of D-RAM available to the client (and server) has increased massively. Laboratory and manufacturing evidence reveals that this trend will continue at least through 1996.

Desktop workstations purchased in 1988 with 1 megabit (Mbit) D-RAM chips were available in 1992 with 4Mbit DRAM chips for the same or lower cost. In 1988, typical desktop workstations contained 1 to 4 megabytes (Mbytes) ofD-RAM. In 1992, these same configurations contain from 4 to 16Mbytes. In 1995, these configurations will use 16Mbit chips and be available with 16 to 64Mbytes for the same price. By 1998—within the life span of many applications being developed today-these configurations will use 64Mbit chips and contain from 64 to 256Mbytes ofD-RAM for the same price.

A revolutionary change is occurring in our capability to provide functionality at the desktop. Most developers cannot generate anywhere near the amount of code necessary to fill a 64Mbyte processor on the desk. Yet applications being built today will be used on desktop processors with this amount of D-RAM. this eliminates the delay that was previously inherent in program switching-that is, program loading and startup. It is now practical to use a multitasking client workstation with several active tasks and to switch regularly among them. Virtual storage is a reality. Workstation D-RAM costs were less than $50 per megabyte in 1992. The cost difference for an additicoal 4 megabytes is only $200. Only one year earlier, short-sighted application designers may have made system design decisions based on a cost of $1000 for 4Mbytes.

The same chip densities used for desktop processors are used in host servers. The typical mainframe computer in 1988 contained from 64 to 256Mbytes of D-RAM. In 1992, 256 to 1,024Mbytes were typical. By 1995, these same host servers will contain 1,024 to 4,096Mbytes ofD-RAM. After 1998, host servers will contain 4,096 to 16,192Mbytes of D-RAM. These quantities are large enough to mandate that we take a completely different view of the way in which software will be built and information will be managed. During the useful life of systems being conceived today, the I/O botrleneck will be eliminated by the capability to access permanent information from D-RAM

We are on the verge of the postscarcity era of processor power. In this era, essentially unlimited computing power will become available. With the client/server model, this processing power is available in every workplace-a fundamental paradigm shift to the information-processing industry and to its customers. We expect to see a significant shakeout in the industry as hardware-only vendors respond to these changes. What will this mean for developers and consumers?

Only manufacturers who offer the lowest prices and who diversify into the software industry will be successful.

Computer technology consumers are now on the verge of a related paradigm shift in the way computer technology affects their business. Only the most effective users can hope to demonstrate real value in a competitive marketplace.

2.3.5 Software Trends

To achieve the benefit of this advance in technology, organizations must choose software that can use it. Traditional development tools, operating systems, character mode user interfaces, and non-SQL-based database technology cannot take advantage of this quantity of D-RAM and the power available from workstation technology.

2.3.5.1 Graphical Screen Designs

Graphical user interfaces (GUis) require large amounts of D-RAM to hold the screen image, pull-down lists, help text, navigation paths, and logic associated with all possible selectable events. Because a GUI enables processing to be selected randomly rather than in the traditional sequential, top-to-bottom order, all possible process logic and GUI management code associated with the image must be available in D-RAM to provide appropriate responses.

GUI functions require subsecond response time. Industry analysis has determined, and our experience confirms, that pull-down lists, button selects, and event invocation should take place within 0.1 second to provide a *suitable* user interface. Suitable means that the user is unaware of the GUI operations but is focused on the business function being performed, This performance is feasibly provided with today's workstations configured with reasonable amounts of $50 per megabyte D-RAM (in 1992) and properly architected applications.

CICS developers do not good GUI developers make.3 GUI application development requires a special mindset. Education, experience, and imagination are prerequisites for moving from the character mode world to the GUI world. Laying out a character mode screen requires that fields are lined up row to row and the screen is not cluttered with too many fields. GUI layout is more difficult, because there are so many ôptions. Colors, pull-down lists, option buttons, text boxes, scrollbars, check boxes, and multiple windows are all layout capabilities. The skills that a layout artist commonly possesses are more appropriate to the task than those which a programmer usually demonstrates.

2.3.5.2 Relational Databases

Another dramatic change in software is in the area of database management. Traditionai file system and database technologies rely on locality of reference for good performance in accessing data. Locality of reference implies that all data needed to satisfy a request is stored physically close together. However, today's business environment requires multikeyed access to rows of information derived from multiple tables. Performance is only possible in these environments when database searches are performed in main storage using extracted keys organized into searchable lists. Physical access to the database is restricted to the selection of rows that satisfy all search criteria.

Relational database technology, using SQL, best meets these criteria.This commonly held view of relational technology is no longer valid. This incorrect view is frequently promulgated by those who have invested their careers in becoming experts in nonrelational technology. Experience indicates that in concert with good development standards and current technology, relational systems perform as well or better than previous technologies. In addition to providing independence of the physical storage from the logical view, SQL processors extract the row descriptors (column values) to separate indexes that are managed in main storage. The search request can be evaluated against the indexes to identify the rows that satisfy all search criteria. Only these identified rows are physically retrieved from external storage.

Standards for use are an important part of a successful implementation of any tool. For example, developers can defeat the effectiveness of SQL in the client/server implementation by coding boolean selection criteria with program logic rather than embedded SQL. Boolean selection criteria retrieves all rows that satisfy the first SELECT condition so that the program logic can be executed to filter unwanted rows. When all the application logic and database processing reside on the same processor, this is a manageable overhead. In a client/server implementation, this causes database selection to operate at the LAN or WAN communication rates rather than at the I/O subsystem rates. Frequently, developers-hoping to reduce the overhead of query eptimization-ı-use the boolean technique for dynamic SQL, with the unfortunate result that performance is dramatically reduced as the additional physical data access time is incurred. It is important to select tools in the client/server world that generate fully qualified SQL SELECT statements.

Relational systems can and do perform, but poor standards of use can defeat them. An example of successful performance, this book has implemented an application, described in Appendix A, that processes more than 400 update transactions per second into a five-table relational database view. This specific example is implemented under DB2,on a midsize ES9000 processor.

2.4 Connectivity

The era of desktop workstations began in 1981 with the introduction of the IBM personal computer (PC). The PC provided early users with the capability to do spreadsheets, word processing, and basic database services for personal data. Within three years, it became clear that high-quality printers, backup tapes, high-capacity disk devices, and software products were too expensive to put on everyone's desktop. LAN technology evolved to solve this problem. Novell is and has been the most successful vendor in the LAl'-J market.

2.4.1 Step I-Workstations Emulate Corporate Systems

In most large organizations, desktop workstations provide personal productivity and some workgroup functions, but host services still provide most other business functions, The lack of desktop real estate encourages the addition of terminal emulation services to the workstation. This emulation capability connects the workstation directly to the corporate systems. The connection was and generally still is provided by a direct connection from the workstation to the host server or its controller. It is possible to use a sub-$5,000 workstation as a $500 dumb terminal.

Connectivity provides the opportunity to move beyond terminal emulation to use the full potential of the workstation. Often the first client/server applications in a large organization use existing mainframe applications. These are usually presentation services-only applications.

2.4.2 Step 2-Adding Servers for Database and Communications

The next step in connectivity is the implementation of specialized servers to provide database and communications services. These servers provide LAN users with a common database for shared applications and with a shared node to connect into the corporate network. The communications servers eliminate the need for extra cabling and workstation hardware to enable terminal emulation. The LAN cabling provides the necessary physical connection, and the communications server provides the necessary controller services.

2.4.3 Step 3--Full-Fledged Client/Server Applications

With its implementation of communications and database servers in place, an organization is ready for the next step up from presentation services-only client/server applications to full-fledged client/server applications. These new applications are built on the architecture defined as part of the system development environment (SOE).

2.5 User Productivity

Personal computer users are accustomed to being in control of their environment. Recently, users have been acclimated to the GUI provided by products such as Windows 3.x, OPEN LOOK, MacOS, and NeXtStep. Productivity is enhanced by the standard look and feel that most applications mnning in these environments provide. A user is trained both to get into applications and to move from function to function in a standard way. Users are accustomed to the availability of context-sensitive help, "friendly" error handling, rapid performance, and flexibility.

Compare the productivity achieved by a financial or budget analyst using a spreadsheet program such as Lotus 1-2-3 or Excel to that achieved when similar functionality is programmed in COBOL on a mainframe. Adding a new variable to an analysis or budget is a trivial task compared to the effort of making functions perform a similar change in the mainframe-based COBOL package. In the first instance, the change is made directly by the user who is familiar with the requirement into a visible model of the problem. In the instance of the mainframe, the change must be made by a programmer, who discusses the requirement with the analyst, attempts to understand the issues, and then tries to make the change using an abstraction of the problem.

The personal computer user makes the change and sees the result. The mainframe programmer must make the change, compile the program, invoke the program, and run the test. If the user understands the request, the implications, and the syntactical requirements, he or she may get it right the first time. Usually, it takes several iterations to actually get it right, often in concert with a frustrated user who tries to explain the real requirement.

We aren't suggesting that all applications can be developed by nonprogrammers using desktop-only tools. However, now that it has become rather easy to build these types of applications on the desktop, it is important for professional IS people to understand the expectations raised in the minds of the end-user community.

2.6 Ways to Improve Performance

Client/server-developed applications may achieve substantially greater performance when compared with traditional workstations or host-only applications.

2.6.1 Offload Work to Server

Database and communications processing are frequently offloaded to a faster server processor. Some applications processing also may be offloaded, particularly for a complex process, which is required by many users. The advantage of offloading is realized when the processing power of the server is significantly greater than that of the client workstation. Shared databases or specialized communications interfaces are best supported by separate processors. Thus, the client workstation is available to handle other client tasks. These advantages are best realized when the client workstation supports multitasking or at least easy and rapid task switching.

2.6.2 Reduce Total Execution Time

Database searches, extensive calculations, and stored procedure execution can be performed in parallel by the server while the client workstation deals directly with the current user needs. Several servers can be used together, each performing a specific function. Servers may be multiprocessors with shared memory, which enables programs to overlap the LAN functions and database search functions. In general, the increased power of the server enables it to perform its functions faster than the client workstation. In order for this approach to reduce the total elapsed time, the additional time required to transmit the request over the network to the server must be less than the saving. High-speed local area network topologies operating at 4, 10, 16, or lOOMbs (megabits per second) provide high-speed communications to manage the extra traffic in less time than the savings realized from the server. The time to transmit the request to the server, execute the request, and transmit the result to the requestor, must be less than the time to perform the entire transaction on the client workstation.

2.6.3 Use a Multitasking Client

As workstation users become more sophisticated, the capability to be simultaneously involved in multiple processes becomes attractive. Independent tasks can be activated to manage communications processes, such as electronic mail, electronic feeds from news media and the stock exchange, and remote data collection (downloading from remote servers). Personal productivity applications, such as word processors, spreadsheets, and presentation graphics, can be active. Several of these applications can be dynamically linked together to provide the desktop information processing environment. Functions such as Dynamic Data Exchange (DDE) and Object Linking and Embedding (OLE) permit including spreadsheets dynamically into word-processed documents. These links can be *hot* so that changes in the spreadsheet cause the word-processed document to be updated, or they can be *cut and paste* so that the current status of the spreadsheet is copied into the word-processed document.

Systems developers appreciate the capability to create, compile, link, and test programs in parallel. The complexity introduced by the integrated CASE environment requires multiple processes to be simultaneously active so the workstation need not be dedicated to a single long-running function. Effective use of modern CASE tools and workstation development products requires a client workstation that supports multitasking.

2.7 How to Reduce Network Traffic

Excessive network traffic is one of the most common causes of poor system performance. Designers must take special care to avoid this potential calamity.

2.7.1 Minimize Network Requests

In the centralized host model, network traffic is reduced to the input and output of presentation screens. In the client/server model, it is possible to introduce significantly more network traffic if detailed consideration is not given to the requestor-server interface.

In the file server model, as implemented by many database products, such as dBASE IV, Fox:Pro, Access, and Paradox, a search is processed in the client workstation. Record-level requests are transmitted to the server, and all filtering is performed on the workstation. This has the effect of causing all rows that cannot be explicitly filtered by primary key selection to be sent to the client workstation for rejection. In a large database, this action can be dramatic. Records that are owned by a client cannot be updated by another client without integrity conflicts. An in-flight transaction might lock records for hours if the client user leaves the workstation without completing the transaction. For this reason, the file server model breaks down when there are many users, or when the database is large and multikey access is required.

However, with the introduction of specific database server products in the client/server implementation, the search request is packaged and sent to the database server for execution. The SQL syntax is very powerful and-when combined with server trigger logic-enables all selection and rejection logic to execute on the server. This approach ensures that the answer set returns only the selected rows and has the effect of reducing the amount of traffic between the server and client on the LAN. (To support the client/server model, dBASE IV, FoxPro, and Paradox products have been retrofitted to be SQL development tools for database servers.)

The performance advantages available from the client/server model of SQL services can be overcome. For example, if by using an unqualified SQL SELECT, all rows satisfying the request are returned to the client for further analysis. Minimally qualified requests that rely on the programmer's logic at the workstation for further selection can be exceedingly dangerous. Quite possibly, 1 million rows from the server can be returned to the client only to be reduced by the client to 10 useful rows. The JOIN function in SQL that causes multiple tables to be logically combined into a single table can be dangerous if users don't understand the operation of the database engine.

A classic problem with dynamic SQL is illustrated by a request to Oracle to JOIN a 10-row table at the client with a 1-million-row table at the server. Depending on the format of the request, either 10 useful rows may be transferred to the client or 1 million rows may be transferred so that the useless 999,990 can be discarded. You might argue that a competent programmer should know better; however, this argument breaks down when the requestor is a business analyst. Business analysts should not be expected to work out the intricacies of SQL syntax. Their tools must protect them from this complexity. (Some DBMSs are now making their optimizers more intelligent to deal with just these cases. So, it is important to look beyond transaction volumes when looking at DBMS engines.) If your business requirement necessitates using these types of dynamic SQL requests, it is important, when creating an SDE, that the architecture definition step selects products that have strong support for query optimization. Products such as Ingres are optimized for this type of request.

Online Transaction Processing (OLTP) in the client/server model requires products that use views, triggers, and stored procedures. Products such as Sybase, Ellipse, and Ingres use these facilities at the host server to perform the join, apply edit logic prior to updates, calculate virtual columns, or perform complex calculations. Wise use of OLTP can significantly reduce the traffic between client and server and use the powerful CPU capabilities of the server. Multiprocessor servers with shared memory available from vendors such as Compaq, Hewlett Packard, and Sun. These enable execution to be divided between processors. CPU-intensive tasks such as query optimization and stored procedures can be separated from the database management processes.

2.7.2 Ease Strain on Network Resources

The use of application and database servers to produce the answer set required for client manipulation will dramatically reduce network traffic. There is no value in moving data to the client when it will be rejected there. The maximum reduction in network overhead is achieved when the only data returned to the client is that necessary to populate the presentation screen. Centralized operation, as implemented in minicomputer and mainframe environments, requires every computer interaction with a user to transfer screen images between the host and the workstation. When the minicomputer or mainframe is located geographically distant from the client workstation, WAN services are invoked to move the screen image. Client/server applications can reduce expensive WAN overhead by using the LAN to provide local communications services between the client workstation and the server. Many client/server applications use mixed LAN and WAN services: some information is managed on the LAN and some on the WAN. Application design must evaluate the requirements of each application to determine the most effective location for application and database servers.

2.7.3 How to Reduce Costs

Cost of operation is always a major design factor. Appropriate choice of technology and allocation of the work to be done can result in dramatic cost reduction.

Each mainframe user requires a certain amount of the expensive mainframe CPU to execute the client portion of the application. Each CICS user uses CPU cycles, disk queues, and D-RAM. These same resources are orders of magnitude cheaper on the workstation. If the same or better functionality can be provided by using the workstation as a client, significant savings can be realized. Frequently existing workstations currently used for personal productivity applications, such as terminal emulation, e-mail, word processing, and spreadsheet work may be used for mission-critical applications. The additional functionality of the client portion of a new application can thus be added without buying a new workstation. In this case, the cost savings of offloading mainframe processing can be substantial.

When you use a communications server on a LAN, each client workstation does not need to contain the hardware and software necessary to connect to the WAN. Communications servers can handle up to 128 clients for the cost of approximately six client communications cards and software. Despite the dramatic reductions in the price of D-RAM, companies will continue to need their existing client workstations. These devices may not be capable of further D-RAM upgrades, or it may not be feasible from a maintenance perspective to upgrade each device. The use of server technology to provide some of the functionality currently provided within a client workstation frees up valuable D-RAM for use by the client applications. This is particularly valuable for DOS-based clients.

The WAN communications functions and LAN services may each be offloaded in certain implementations. The use of WAN communications servers has the additional advantage of providing greater functionality from the dedicated communications server.

2.8 Vendor Independence

If client and server functionality is clearly split and standards-based access is used, there can be considerable vendor independence among application components. Most organizations use more expensive and more reliable workstations from a mainstream vendor such as Compaq, IBM, Apple, Sun, or Hewlett-Packard for their servers. Other organizations view client workstation technology as a commodity and select lower-priced and possibly less-reliable vendor equipment. The mainstream vendors have realized this trend and are providing competitively priced client workstations. Each of the mainstream vendors reduced its prices by at least 65 percent between 1991-93, primarily in response to an erosion of market share for client workstations.The controversy over whether to move from offering a high-priced but best-quality product line to offering a more competitive commodity traumatized the industry in 1991, forcing Compaq to choose between retaining its founder as CEO or replacing him with a more fiscally aware upstart.

The resulting shakeout in the industry has significantly reduced the number of vendors and makes the use of traditionally low priced clones very risky. Hardware can generally be supported by third-party engineers, but software compatibility is a serious concern as organizations find they are unable to install and run new products.

The careful use of SQL and RPC requests enable database servers and application services to be used without regard to the vendor of the database engine or the application services platform. As noted previously, the operating system and hardware platform of the server can be kept totally independent of the client platform through the proper use of an SDE. However, use of these types of technologies can vastly complicate the development process.

An excellent example of this independence is the movement of products such as FoxPro and Paradox to use client services to invoke, through SQL, the server functions provided by Sybase SQL Server. A recent survey of client development products that support the Sybase SQL Server product identified 129 products. This is a result of the openness of the API provided by Sybase. Oracle also has provided access to its API, and several vendors-notably Concentric Data Systems, SQL Solutions, and Dataliase=-have developed front-end products for use with Oracle. ASK also has realized the importance of open access to buyers and is working with vendors such as Fox and PowerBuilder to port their front ends in support of the Ingres database engine.

An application developed to run in a single PC or file server mode can be migrated without modification to a client/server implementation using a database server. Sybase, Oracle, and Ingres execute transparently under Windows NT, OS/2, or UNIX on many hardware platforms. With some design care, the server platform identity can be transparent to the client user or developer. Despite this exciting oppoıtunity, progranuners or manufacturers often eliminate this transparency by incorporating UNIX-, Windows NT-, or OS/2-specific features into the implementation. Although FoxPro can work with SQL and Sybase, the default Xbase format for database access does not use SQL and therefore does not offer this independence. To take advantage of this platform transparency, organizations must institute standards into their development practices.

2.9 Faster Delivery of Systems

Some software development and systems integration vendors have had considerable success using client/server platforms for the development of systems targeted completely for mainframe execution. These developer workstations are often the first true client/server applications implemented by many organizations. The workstation environment, powerful multitasking CPU availability, single-user databases, and integrated testing tools all combine to provide the developer with considerable productivity improvements in a lower-cost environment. Our analysis shows that organizations that measure the "real" cost of mainframe computing will cost justify workstation development environments in 3 to 12 months.

Client/server application development shows considerable productivity improvement when the software is implemented within an SDE. As previously noted, organizational standards-based development provides the basis for object-oriented development techniques and considerable code reuse. This is particularly relevant in the client/server model, because some natural structuring takes place with the division of functionality between the client and server environments. Reuse of the server application functionality, database, and network services is transparent and almost automatic. Because the applications are built with little regard to standard front-end functionality, ~any features are part of the standard GUI and are automatically reused.

2.9.1 Smaller and Simpler Problems

Client/server applications frequently are involved with data creation or data analysis. In such applications, the functionality is personal to a single user or a few users. These applications frequently can be created using standard desktop products with minimal functionality. For example, data may be captured directly into a form built with a forms development tool, edited by a word processor, and sent on through the e-mail system to a records management application. In the back end, data may be downloaded to a workstation for spreadsheet analysis.

2.9.2 Less Bureaucratic Inertia

Mainframes provide the stable, reliable environment that is/ desirable and necessary for production execution. This same stability is the bane of developers who require rapid changes to their test environments. The workstation environment is preferable because it is personal and responds to the user's priorities. Developers can make changes at their own pace and then deal with the mainframe bureaucracy if and when the application goes into production in the mainframe environment.

Many users typically run applications on the mainframe. Changes made to such applications affect all their users. In some instances, the entire mainframe may be unavailable during the implementation of a new application. Network reconfiguration, database utilities, application definition, and system software maintenance all can impacts users beyond those specifically involved in a change. It is awkward to migrate only a portion of the users from the previous implementation to the new one. Typically, it is all or none of the users who must upgrade. This change process requires thorough and all-encompassing tests and careful control over the move to production.

The client/şerver environment provides more flexibility for phased implementation of the new production environment. The application is replicated at many different locations so the users may implement the new software individually rather than all at once. This environment adds the additional and significant complication of multiple updates. New products are now available from vendors such as Synchrony, Hewlett-Packard, and IBM that automate and control this function.

Workgroup client/server applications frequently are used-by fewer users. These users can be directly supported by the developer immediately after implementation. Corrections can be made and reimplemented more readily. This is not to suggest that in the client/server world change and production control procedures are not necessary, only that they can be less onerous for workgroup applications. Remote LAN management will be required for enterprise applications implemented throughout the corporation. Only in this way will support equivalent to that available today for host-based applications be available to remote client/server users.

--- 3 ---

Components of Client/Server Applications

--- The Client ---

3.1 Executive Summary

The client in the client/server model is the desktop workstation. Any workstation that is used by a single user is a client. The same workstation, when shared simultaneously by multiple users, is a server. An Apple Macintosh SE, an IBM PS/2 Model 30, an ALR 386/220, a Compaq SystemPro, an NCD X-Terminal, a Sun Sparcstation, a DECstation 5000-all are used somewhere as a clien.f workstation. There is no specific technological characteristic of a client.

During the past IO years, workstation performance improved dramatically. For the same cost, workstation CPU performance increased by 50 times, main memory has increased by 25 times, and permanent disk storage has increased by 30 times. This growth in power allows much more sophisticated applications to be run from the desktop.

Communications and network speeds have improved equally in the last 10 years. In 1984, the performance and reliability ofremote file, database, and print services were inadequate to support business applications. With the advent of high-speed local and wide area networks (LANs and WANs), networking protocols, digital switches, and fiber-optic cabling, both peıformance and reliability improved substantially. It is now practical to use these remote services as part of a critical business application.

The client workstation may use the DOS, Windows, Windows NT, OS/2, MacOS (also referred to as System 7), or UNIX operating system. The client workstation frequently provides personal productivity functions, such as word processing, which use only the hardware and software resident right on the workstation. When the client workstation is connected to a LAN, it has access to the services provided by the network operating system (NOS) in addition to those provided by the client workstation. The workstation may load software and save word-processed documents from a server and therefore use the file server functions provided through the NOS. It also can print to a remote printer through the NOS. The client workstation may be used as a terminal to access applications resident on a host minicomputer or mainframe processor. This enables the single workstation to replace the terminal, as well as provide client workstation functionality.

In a client/server application, :functions are provided by a combination of resources using both the client workstation processor and the server processor. For example, a database server provides data in response to an SQL request issued by the client application. Local processing by the client might calculate the invoice amount and format the response to the workstation screen.

Client workstations can provide business :functions using a mixture of personal productivity products in conjunction with a custom application. For example, a document created by a word processor can include input from a spreadsheet program and the invoice data created by the client/server application. The capability to cut and paste input from several different sources is one of the most powerful aspects of a client workstation. It provides the end user with tools to create new applications-without the need to go to professional programmers for assistance.

3.2 The Role of the Client

In the client/server model, the client is primarily a consumer of services provided by one or more server processors. The model provides a clear separation of functions based on the idea of servers acting as service providers responding to requests from clients. It is important to understand that a workstation can operate as a client in some instances while acting as a server in other instances. For example, in a LAN Manager environment, a workstation might act as a client for one user while simultaneously acting as a print server for many users. This chapter discusses the client functions.

The client almost always provides presentation services. User input and final output, if any, are presented at the client workstation. Current technology provides cost effective support for a graphical user interface (GUI). This book recommends that all new applications, with direct interaction by a human, be developed using a GUI. The windowing environment enables the client user to be involved in several simultaneous sessions: Such :functions as word processing, spreadsheet, e-mail, and presentation graphics=-in addition to the custom applications built by the organization-can be active simultaneously. Windows 3.x and Mac System 7 do not support true multitasking; thus, only one task at a time can safely be engaged in a communications session. Windows NT, OS/2, and UNIX are preemptive multitasking operating systems and thus will support any number of active communications sessions.

Facilities such as Dynamic Data Exchange (DDE), Object Level Embedding (OLE), and Communicating Object Request Broker Architecture (CORBA), which are discussed later in this chapter, provide support for cut-and-paste operations between word processors, databases, spreadsheets, and graphics in a windowing environment. Beyond this, a selectable set of tasks may be performed at the client. In fact, the client workstation can be both client and server when all information and logic pertinent to a request is resident and operates within the client workstation.

Software to support specific functions-for example, field edits, con-text-sensitive help, navigation, training, personal data storage, and manipulation-frequently executes on the client workstation. *All* these functions use the GUI and windowing functionality. Additional business logic for calculations, selection, and analysiscan reside on the client workstation.

A client workstation uses a local operating system to host both basic services sad the network operating system interfaces. This operating system may be the same or different from that of the server. Most personal computer users today use DOS or Windows 3.x: as their client operating system, because current uses are primarily personal productivity applications-e- not ones requiring a client/server.

Those users running client/server applications from DOS or Windows typically run only a single business process at a time. However, the demand to use these familiar operating systems is driving the development of client/server tools such as Powerlsuilder for Windows, and new multitasking versions of Windows (such as Win<lows NT, Windows 4-expected to be available in late 1994-and Cairo, expected in late 1995). Fortunately, the advent of products such as Digitalk's Parts and Pare Place's Visual Works provide development tools that are equally happy running in the Windows 3 .x or OS/2, UNIX, and Windows NT worlds.

Because UNIX and OS/2 have lacked the familiar personal productivity tools such as word processors, e-mail, spreadsheets, presentation graphics, and database management systems, DOS and Windows have become the client operating systems of choice. Until recently, few personal productivity applications for OS/2 and UNIX were in place, and client/server requirements that dictate OS/2 and UNIX were not evident. Now, improvements in the capability of these operating systems to run personal productivity applications, and increased user needs for high reliability or for multitasking has increased the popularity of OS/2, X-Terminals, and UNIX. Native execunion of Windows 3.1 applications under Windows NT, OS/2, and many UNIX implementations offers the best of all worlds for the desktop user: reliability and functionality, The current availability of OS/2 Version 2.1, UNIX, and Windows NT with integrated support for DOS, Windows 3.x, and X-Windows-as well as support for multitasking in a reliable environment-is a continuing reason for making these the client operating systems the choice for developing business critical client/server applications. As noted, the dramatic reduction in processor and D-RAM costs make the extra resources required for OS/2, UNIX, and Windows NT minimal. Finally, the software licensing costs for OS/2 2.x, UNIX from Sun and USL are comparable to that for DOS and Windows 3.x.

UNIX supports many of the most familiar personal computer app-lications, such as Lotus 1-2-3, WordPerfect, and dBASE IV. This fact-coupled with the availability of low-cost, high-performance RISC processors-is making UNIX a strong contender as a client for client/server applications. During 1994-1995, it is expected that multitasking desktops provided by Windows NT, Windows 4.x, UNIX, and OS/2 will become the operating systems of choice for clients in a client/server environment. Selection between Windows versions, UNIX, and OS/2 will be made on the basis of cost performance rather than functionality. Previously purchased PC limitations will encourage many organizations to remain with Windows 4 and OS/2 rather than Windows NT or UNIX, which might require new hardware acquisitions. OSF/1 (a commercial-grade UNIX) is now available for the Intel platform and is causing organizations to reexamine the use of UNIX on the PC. The current licensing costs for OS/2 may give OS/2 the edge unless OSF/1 costs are much less than current UNIX licenses.

The Common Open Software Environment (COSE) group of UNIX kernel vendors has agreed on a common set of API's for most UNIX services. This allows application developers to build one application for all platforms. This will serve to expand the number of applications that will run across the various UNIX platforms. In turn, this will increase the use of UNIX on the desktop and subsequently reduce the per-seat cost.

Windows 3.x is by far the dominant GUI and even with its single tasking limitations, it is a leading client operating system candidate for client/server applications. Microsoft's Windows 4, the planned upgrade for Windows 3.x, is discussed more fully in Appendix B. It will provide a client platform that can better use the capabilities of the new generation ofintel processors while continuing to provide the GUI and API's of Windows 3.x. This operating system is likely to gain a significant share of the client user base in 1995. The complexity and resource requirements of Windows NT suggest it will not displace many Windows desktops prior to the availability of Windows 4.

In terms of known "wild cards" for the client OS, IBM and Apple have formed an alliance with Motorola to develop a new client operating system in a venture known now as Taligent. This new OS is based on AIX, OS/2, and Mac System 7. The result should be a client platform with the ease of use interface of Mac System 7, and the functionality and connectivity of AIX and OS/2. (This subject is discussed more fully in Chapter 10.) This initiative will bear fruit during 1994 and will compete during 1995 for the role of preferred client platform. Microsoft's competitor in this market, currently code named Cairo, will reach the market in late 1995 and will compete during 1996 for the multitasking desktop market.

With the uncertainty surrounding the operating system alternatives, it is important that all development be done with an SDE that isolates the operating system from the application. Then, if operating system changes are warranted the applications should be able to port without any impact beyond recompilation.

3.3 Client Services

The ideal client/server platform operates in an open systems environment using a requester-server discipline that is based on well-defined standards. This enables multiple hardware and software platforms to interact. When the standard requester-server discipline is adhered to, servers may grow and change their operating system and hardware platforms without changing the client applications. Clients can be entry-level Intel 386SX machines or very powerful RISC-based workstations, and run the same application issuing the same requests for service as long as the standard requester-server discipline is adhered to. Traditional host applications that use the client for presentation services operate only by sending and receiving a character data stream to and from a server. All application logic resides on the server. This is the manner in which many organizations use workstation technology today. The expensive mainframe CPU is being used to handle functions that are much more economically provided by the workstation.

First-generation client/server applications using software such as Easel enable the input and output data streams to be reformatted at the client without changes to the host applications. They use an API that defines the data stream format. Easel uses the IBM-defined Extended High Level Language Application Program Interface

(EHLLAPI). GUI front ends may add additional functionality, such as the capability to select items for input from a list, selectively use color, or merge other data into the presentation without changing the host application.

An example of this form of client is an application developed for the emergency command and control services required by E911 dispatch applications. This computer application supports calls to the 91 l emergency telephone number and dispatches fire, police, ambulance, or emergency vehicles to an incident. This application traditionally has been implemented on a fault-tolerant minicomputer with access provided from a character mode dumb terminal. The information is displayed in list form, and the operator can move the cursor to an item on the list for selection or rekey the data for input. Prior implementations of this application handled the address of the caller by displaying it on the screen as a text field.

In the client/server implementation of this system, the workstation user deals only with a GUI. The workstation plots this address onto a map that in turn displays the location of the fire. In addition, the locations of all fire stations and vehicles are plotted on the map. The dispatch operator can see at a glance the entire status of fire support close to the fire. Previous implementations of this application displayed lists of optional fire vehicles. From this list, the operator keyed in a selected vehicle. The GUI front end, however, enables the vehicles to be shown in a window and selected by using a mouse pointer. This not only reduces the cost of execution but can significantly reduce errors, increase productivity, and reduce stress experienced by the dispatch operator.

Gills enable users to be more productive with less training, because the interface is more intuitive. Several studies comparing the productivity and learning curve for users of GUI applications versus traditional character mode applications have demonstrated improvements of greater than 200 percent.

The functionality of the client process can be further extended at the client by adding logic that is not implemented in the host server application. Local editing, automatic data entry, help capabilities, and other logic processes can be added in front of the existing host server application. If many errors are detected at the client, or functions such as online help are completely off loaded, the workload of the host server decreases. There is an opportunity to provide extensive interactive help and training integrated into a client/server application using only the services of the client workstation and NOS.

One example of this functionality is shown by an application developed for the state of Hawaii. To determine welfare eligibility, state employees conduct an extensive analysis of each applicant's personal situation. The process of capturing this information is time-consuming and stressful for the case worker and the applicant. Hawaii addressed this requirement by using an "unattended" kiosk for the interview-an interactive video unit provides the questions and displays a set of possible responses. Users enter responses on a touch screen and can respond to the questions at their own rate. The case worker is not tied up with the mechanics of filling out the questionnaire, and the state has the opportunity through the interactive video to ensure that applicants are aware of all their rights and responsibilities. The case worker and applicant review the application after it is completed. The existing computer system captures and edits the data and performs the final eligibility determination. A dramatically different and more effective user interface is provided while preserving much of the investment in existing computer systems.

Completion of multipart forms often involves redundant data entry into multiple computer systems or applications. Collecting this data at the source or into a common data entry function and distributing it to the other data entry functions can reduce costs and errors. Ideally, the information is entered by the individual or process responsible for the data creation. This enables the individual with the knowledge to make corrections and to do so immediately. The workgroup LAN server captures the data and stores it. When a business process defined to capture data from one copy of the form is invoked, the stored data is automatically merged into the form. This is updated, by the user, with additional data that is now available. In this manner, data is keyed only once and every business process uses the same data. Information is made available immediately after capture and can be distributed electronically to all authorized users.

It is possible to make fundamental changes in the business process, using a Business Process Reengineering (BPR) methodology and client/server computing. One such example uses electronic imaging. Many firms have found that it pays to put a series of steps that formerly involved different people handling each step, onto the shoulders of a single "case worker." One insurance company, for example, estimated that it took 22 days to approve a policy, during which time the papers were worked on for only 17 minutes. The remainder of the time was spent shuffling papers between specialists=from credit-checkers to actuaries to salespeople and back. By enabling everyone in an organization to share information more or less instantly, new technology highlights the fact that most insurance policies never need be seen by most of these Specialists. As long as specialists can be consulted quickly when needed, the vast majority of policies can be handled by a single person. Mutual Benefit Life used such a procedure to boost productivity among clerical staff by 60 percent.[1]

Another commonly used technique to leverage the power and ease of use of the workstation is provided by tools, such as Trinzic's Forest & Trees. These tools provide easy-to-use facilities to manipulate data either stored on the existing host databases or downloaded to local servers. This technique of "data mining" through the use of powerful developer tools to provide rapid development of new management decision support functions, portends the future for systems development. Future developers will be *knowledge workers* -technologists with an equally strong business understanding using tools that are intuitive and powerful. Data will be provided to the workstation user in a form consistent with his or her business understanding.

Why is workstation technology so effective? It supports the new business paradigm of employee empowerment. It provides the windowing capabilities to simultaneously access and display all information necessary to complete the business process. The capability of powerful workstation technology to recommend and make decisions based on historical precedent can dramatically reduce cost and improve service by shortening the decision-making cycle.

3.4 Request for Service

Client workstations request services from the attached server. Whether this server is in fact the same processor or a network processor, the application format of the request is the same. NOS software translates or adds the specifics required by the targeted requester to the application request.

Interprocess communication (IPC) is the generic term used to describe communication between running processes. In the client/server model, these processes might be on the same computer, across the LAN, or across the WAN.

The most basic service provided by the NOS is *redirection.* This service intercepts client workstation operating system calls and redirects them to the server operating system. In this way, requests for disk directories, disk files, printers, printer queues, serial devices, application programs, and named pipes are trapped by the redirection software and redirected (over the LAN) to the correct server location. It is still possible for some of these services to be provided by the client workstation. The local disk drives may be labeled A: and C: and the remote drives labeled D:, E:, and F:.

How does redirection work?

Any request for drive A: or C: is passed through to the local file system by the redirection software. Requests for other drives are passed to the server operating system. Printers are accessed through virtual serial and parallel ports defined by the NOS redirector software.

The NOS requester software constructs the remote procedure call (RPC) to include the API call to the NOS server. The NOS server then processes the request as if it were executed locally and ships the response back to the application.

Novell commercialized this redirector concept for the Intel and MS-DOS platforms, and it has been adopted by ali NOS and UNIX network file system (NFS) vendors. The simplicity of executing standard calls to a virtual network of services is its main advantage.

3.4.1 Remote Procedure Cail (RPC)

Over the years, good programmers have developed modular code using structured techniques and subroutine logic. Today, developers want subroutines to be stored as a named objects "somewhere" and made available to everyone with the right to use them. Remote procedure calls (RPCs) provide this capability. RPCs standardize the way programmers must write calls, so that remote procedures can recognize and respond correctly.

If an application issues a functional request and this request is embedded in an RPC, the requested function can be located anywhere in the enterprise that the caller is authorized to access. The RPC facility provides for the invocation and execution of requests from processors running different operating systems and using hardware platforms different from that of the caller. Many RPCs also provide data translation services. The call causes dynamic translation of data between processors with different physical data storage formats. These standards are evolving and being adopted by the industry

3.4.2 Fax/Print Services

The NOS enables the client to generate print requests even when the printer is busy. These are redirected by the NOS redirector software and managed by the print server queue manager. The client workstation can view the status of the print queues at any time. Many print servers notify the client workstation when the print request is completed. Fax services are made available in exactly the same manner as print servers, with the same requester server interface and notification made available.

3.4.3 Window Services

A client workstation may have several windows open on-screen at any time. The capability to activate, view, move, size, or hide a particular window is provided by the window services of the client operating system. These services are essential in a client/server implementation, because they interact with message services provided to notify the user of events that occur on a server. Application programs are written with no sensitivity to the windowing. Each application is written with the assumption that it has a virtual screen. This virtual screen can be an arbitrary size and can even be larger than the physical screen.

The application, using GUI software, places data into the virtual screen, and the windowing services handle placement and manipulation of the application window. This greatly simplifies application development, because there is no need for the developer to build or manage the windowing services. The client user is totally in control of his or her desktop and can give priority to the most important tasks at hand simply by positioning the window of interest to the "front and center." The NOS provides software on the client workstation to manage the creation of pop-up windows that display alerts generated from remote servers. E-mail receipt, print complete, Fax available, and application termination are examples of alerts that might generate a pop-up window .to notify the client user.

3.4.4 Remote Boot Services

Some applications operate well on workstations without any local disk storage; X-terminals and workstations used in secure locations are examples. The client workstation must provide sufficient software burned into erasable programmable read-only memory (E-PROM) to start the initial program load (IPL)-that is, boot-process. E-PROM is included in all workstations to hold the Basic Input/Output System (BIOS) services. This mini-operating system is powerful enough to load the remote software that provides the remainingservices and applications functions to the client workstation or X-terminal.

3.4.5 Other Remote Services

Applications can be invoked from the client to execute remotely on a server. Backup services are an example of services that might be remotely invoked from a client workstation. Business functions such as downloading data from a host or checking a list of stock prices might also be invoked locally to run remotely.

Software is provided by the NOS to run on the client workstation to initiate these remote applications.

Mobile computing is increasingly being used to remain functional while out of the office. With appropriate architectural forethought, applications can be built to operate effectively from the office LAN or the remote laptop. Current technology supports full-powered workstations with the capability for GUI applications consistent with the desktop implementation. The IPC protocol of choice for mobile access is TCP/IP based.

3.4.6 Utility Services

The operating system provides local functions such as copy, move, edit, compare, and help that execute on the client workstation.

3.4.7 Message Services

Messages can be sent and received synchronously to or from the network. The message services provide the buffering, scheduling, and arbitration services to support this function.

3.4.8 Network Services

The client workstation communicates with the network through a set of services and APis that create, send, receive, and format network messages. These services provide support for communications protocols, such as NetBIOS, IPX, TCP/IP, APPC, Ethernet, Token Ring, FDDI, X.25, and SNA. These are more fully described in Chapter 5, "Components of Client/Server Applications-Connectivity."

3.4.9 Application Services

In addition to the remote execution services that the NOS provides, custom applications will use their own APis embedded in an RPC to invoke specialized services from a remote server.

3.4.10 Database Services

Database requests are made using the SQL syntax. SQL is an industry standard language supported by many vendors. Because the language uses a standard form, the same application may be run on multiple platforms. There are syntactical differences and product extensions available from most vendors. These are provided to improve developer productivity and system performance and should be carefully evaluated to determine whether their uses are worth the incompatibility implied by using proprietary components. Using unique features may prevent the use of another vendor's products in a larger or smaller site. Certain extensions, such as stored procedures, are evolving into *de facto* standards.The use of stored procedures is often a way of avoiding programmer use of proprietary extensions needed for performance. A clear understanding, by the technical architects on the project, of where the standards are going is an important component of the SDE standards for the project.

3.4.11 Network Management Services-Alerts

Most network interface cards (NICs) can generate alerts to signify detected errors and perhaps to signify messages sent and received. These alerts are valuable in remote LAN management to enable early detection of failures. Because many errors are transient at first, simple remote detection may allow problems to be resolved before they become critical. Applications may also generate alerts to signify real or potential problems, Certain error conditions indicate that important procedures are not being followed. Application program failure may occur because current versions of software are not being used.

Support for a remote client workstation may be greatly simplified if alerts are generated by the applications. This should be part of every standard SDE. Many alert situations can be generated automatically from standard code without the involvement of the application developer.

3.4.12 Dynamic Data Exchange (DDE)

DDE is a feature of Windows 3.x and OS/2 Presentation Manager that enables users to pass data between applications from different vendors through support for common APIs. For example, a charting package can be linked to a database to provide the latest chart data whenever the chart is referenced.

3.4.13 Object Linking and Embedding (OLE)

OLE is an extension to DDE that enables objects to be created with the object components software *aware.* Aware means that a reference to the object or one of its components automatically launches the appropriate software to manipulate the data. For example, a document created with a word processor may include an image created by a graphics package. The image can be converted to the internal graphics form of the word processor, such as WPG form for WordPerfect. With OLE, the image can be included in its original form within the document object; whenever the image is selected or highlighted, the graphics package will take control to manipulate the image. Activation of the software is totally transparent to the users as they navigate through the document.

Currently with OLE, one software package accesses data created from another through the use of a *viewer* or *launcher.* These viewers and launchers must be custom built for every application. With the viewer, users can see data from one software package while they are tunning another package. Launchers invoke the software package that created the data and thus provide the full functionality of the launched software.Both these techniques require the user to be aware of the difference between data sources. DDE and OLE provide a substantial advantage: any DDE- or OLE-enabled application can use any software that supports these data interchange APIs. An e-mail application will be able to attach any number of components into the mail object without the need to provide custom viewers or launchers.

Not all Windows applications support OLE, but Microsoft has released its OLE 2.0 software development kit (SDK). The toolkit greatly simplifies OLE integration into third-party, developed applications. Organizations wanting to create a consistent desktop are beginning to use the OLE SDK as part of custom applications.

OLE 2.0 extends OLE capabilities to enable a group of data to be defined as an object and saved into a database. This object can then be dragged and dropped into other applications and edited without the need to switch back to the application which created it. This provides a more seamless interface for the user. In OLE 1.x, double-clicking a Lotus 1-2-3 for Windows spreadsheet embedded in a Microsoft Word for Windows document launches 1-2-3 and opens the document in a 1-2-3 window. Under OLE 2.0, the active window (Word's) menu and toolbar change to that of 1-2-3. The user deals only with the object, with no need to be aware of the multiple software being loaded

3.4.14 Common Object Request Broker Architecture (CORDA)

CO:RBAis a specification from the Object Management Group (OMG), a UNIX vendor consortium. OLE focuses on data sharing between applications on a single desktop, and CORBA addresses cross-platform data transfer and the process of moving objects over networks. CORBA support enables Windows and UNIX clients to share objects. A word processor operating on a Windows desktop can include graphics generated from a UNIX workstation.

3.5 Enterprise View

It is important for application designers and developers to understand and remember that the user view of the system is through the client workstation. Whatever technological miracles are performed at the server, a poor design or implementation at the client on the desktop still result in unfavorable user perception of the entire application!

--- 4 ---

Components of Client/Server Applications

--- The Server ---

4.1 Executive Summary

The server is a multiuser computer. There is no special hardware requirement that turns a computer into a server. The hardware platform should be selected based on application demands and economics. Servers for client/server applications work best when they are configured with an operating system that supports shared memory, application isolation, and preemptive multitasking. An operating system with preemptive multitasking enables a higher priority task to preempt or take control of the processor from a currently executing, lower priority task.

The server provides and controls shared access to server resources. Applications on a server must be isolated from each other so that an error in one cannot damage another. Preemptive multitasking ensures that no single task can take over all the resources of the server and prevent other tasks from providing service. There must be a means of defining the relative priority of the tasks on the server. These requirements are specific to the client/server implementation and not to the file server implementation. Because file servers execute only the single task of file service, they can operate in a more limited operating environment without the need for application isolation and preemptive multitasking.

The traditional minicomputer and mainframe hosts have acted as de facto enterprise servers for the network of terminals they support. Because the only functionality available to the terminal user is through the host, personal productivity data as well as corporate systems information is stored on this host server. Network services, application services, and database services are provided centrally from the host server.

Many organizations download data from legacy enterprise servers for local manipulation at workstations. In the client/server model, the definition of server will continue to include these functions, perhaps still implemented on the same or similar platforms. Moreover, the advent of open systems based servers is facilitating the placement of services on many different platforms. Client/server computing is a phenomenon that developed from the ground up. Remote workgroups have needed to share expensive resources and have connected their desktop workstations into local area networks (LANs). LANs have grown until they are pervasive in the organization. However, frequently(similar to parking lots) they are isolated one from the other.

Many organizations have integrated the functionality of their dumb terminals their desktop workstations to support character mode, host-based applications from single workstation. The next wave of client/server computing is occurring now, as organizations of the mid-1990s begin to use the cheaper and more available processing power of the workstation as part of their enterprise systems.

The Novell Network Operating System (NOS), NetWare, is the most widely installed LAN NOS. It provides the premier file and print server support. However, a limitation of NetWare for the needs of reliable client/server applications has been the requirement for an additional separate processor running as a database server. The availability of database server software-from companies such as Sybase and Oracle--to run on the NetWare server, is helping to diffuse this limitation. With the release of Novell 4.x, Netware supports an enterprise LAN (that is, a thousand internetworked devices) with better support for Directory Services and TCP/IP intemetworking.

DEC demonstrated the Alpha AXP processor running Processor-Independent NetWare in native mode at the PC Expo exhibit in June 1993. HP, Sun, and other vendors developing NetWare on RISC-based systems announced shipment of developer kits for availability in early 1994. Native NetWare for RISC is scheduled for availability in late 1994. This will provide scalability for existing Netware users who run out of capacity on their Intel platforms.

Banyan VINES provides the competitive product to Novell 4.x for enterprise LANs. Directory services are provided in VINES through a feature called StreetTalk. VINES 5.5 provides excellent WAN connectivity and is very popular among customers with a heterogeneous mainframe and minicomputer enterprise. However, it suffers from a weak support for file and printer sharing and a general lack of application package support. Banyan's Enterprise Network Services (ENS) with StreetTalk provides the best Directory Services implementation today. StreetTalk enables users to log into the network rather than to a server. This single logon ID enables access to all authorized servers anywhere in the network. Banyan made ENS available for Netware 3.11 and plans to make it available for Netware 4.x and Microsoft's Windows NT Advanced Server.

Microsoft's LAN Manager NOS and its several derivatives-including IBM Lan Server, HP LAN Manager/OX and DEC Pathworks-provide file and printer services but with less functionality, and more user complexity, than Novell's NetWare. The operating systems that support LAN Manager provide the necessary shared memory, protected memory, and preemptive multitasking services necessary for reliable client/server computing. They provide this support by operating natively with the OS/2, UNIX, VMS, and MVS operating systems. These operating systems all provide these services as part of their base functionality. The scalability of the platforms provides a real advantage for organizations building client/server, and not just file server, applications.

The lack of reasonable directory services restricts LAN Manager from the enterprise LAN role today. Microsoft has just released Advanced Server, the Windows NT version of LAN Manager. This provides a much stronger Intel platform than LAN Manager. In conjunction with the Banyan ENS, Advanced Server is a strong competitor to Novell's NetWare as the preferred NOS.

Network File System (NFS) is the standard UNIX support for shared files and printers. NFS provides another option for file and print services to client workstations with access to a UNIX server. PC NFS is the PC product that runs on the client and provides connectivity to the NFS file services under UNIX. NFS with TCP/IP provides the additional advantage of easy-to-use support for remote files and printers.

Novell and NFS can interoperate effectively because of the increasing support for TCP/IP as a LAN and WAN protocol. Recent announcements by IBM and Microsoft of alliances with Novell and Banyan promise a future in which all of the features of each NOS will be selectively available to everyone. Until these products improve their capability to work together, organizations still have the challenge of determining which NOS to select. Most will choose to use NetWare plus Windows clients with OS/2, UNIX, VMS, or MVS servers for their client/server applications. There will be a significant increase during 1994-95 in the use of NFS based servers with support now available on all major UNIX platforms as well as OS/2, MVS, and VMS.

There is no preeminent hardware technology for the server. The primary characteristic of the server is its support for multiple simultaneous client requests for service. Therefore, the server must provide multitasking support and shared memory services. High-end Intel, RISC (including Sun SPARC, IBM/Motorola PowerPC, HP PA RISC, SGI MIPS, and DEC Alpha), IBM System/370, and DEC VAX processors are all candidates for the server platform. The server is responsible for managing the server-requester interface so that an individual client request response is synchronized and directed back only to the client requester. This implies both security when authorizing access to a service and integrity of the response to the request.

With object-oriented technology (OOT) increasingly used to build operating systems and development environments, servers are becoming ubiquitous (anything, anywhere, and anytime) and transparent in technology and location to the user and developer. NeXtStep provides the only production ready model of what will be the dominant developer model in 1995 and beyond. Sun's DOE implementation of the OMG defined CORBA standards provides a view of the future role of the object server. This is the first implementation of the vision of the original OOT scientists. The future promises applications assembled from object repositories containing the intellectual properry of a business combined with commercial objects made available by OOT developers executing on servers somewhere.

4.2 The Role of the Server

Servers provide application, file, database, print, fax, image, communications, security, systems, and network management services. These are each described in some detail in the following sections.

It is important to understand that a server is an architectural concept, not a physical implementation description. Client and server functions can be provided by the same physical device. With the movement toward peer computing, every device will potentially operate as a client and server in response to requests for service.

Application servers provide business functionality to support the operation of the client workstation. In the client/server model these services can be provided for an entire or partial business function invoked through an InterProcess Communication (IPC) request for service. Either message-based requests (*a la* OLTP) or RPCs can be used. A collection of application servers may work in concert to provide an entire business function. For example, in a payroll system the employee information may be managed by one application server, earnings calculated by another application server, and deductions calculated by a third application server. These servers may run different operating systems on various hardware platforms and may use different database servers. The client application invokes these services without consideration of the technology or geographic location of the various servers. Object technology provides the technical basis for the application server, and widespread acceptance of the CORBA standards is ensuring the viability of this trend. File servers provide record level data services to nondatabase applications.

Space for storage is allocated, and free space is managed by the file server. Catalog functions are provided by the file server to support file naming and directory structure. Filename maximum length ranges from 8 to 256 characters, depending on the particular server operating system support. Stored programs are typically loaded from a file server for execution on a client or host server platform.

Database servers are managed by a database engine such as Sybase, IBM, Ingres, Informix, or Oracle. The file server provides the initial space, and the database engine allocates space for tables within the space provided by the file server. These host services are responsible for providing the specialized data services required of a database product-automatic backout and recovery after power, hardware, or software failure, space management within the file, database reorganization, record locking, deadlock detection, and management. Print servers provide support to receive client documents, queue them for printing, prioritize them, and execute the specific print driver logic required for the selected printer. The print server software must have the necessary logic to support the unique characteristics of each printer. Effective print server support will include error recovery for jams and operator notification of errors with instructions for restart.

Fax servers provide support similar to that provided by print servers. In addition, fax servers queue up outgoing faxes for later distribution when communications charges are lower. Because fax documents are distributed in compressed form using either Group III or Group IV compression, the fax server must be capable of dynamically compressing and decompressing documents for distribution, printing, and display. This operation is usually done through the addition of a fax card to the server. If faxing is rare, the software support for the compression and decompression options can be used. Image servers operate in a manner similar to fax servers.

Communications servers provide support for wide area network (WAN) communications. This support typically includes support for a subset of IBM System Network Architecture (SNA), asynchronous protocols, X.25, ISDN, TCP/IP, OSI, and LAN-to-LAN NetBIOS communication protocols. In the Novell NetWare implementation, Gateway Communications provides a leading communications product. In the LAN Server and LAN Manager environments, OS/2 communications server products are available from IBM and DCA In the Banyan VINES environment, the addition of DCA products to VINES provides support for SNA connectivity. UNIX servess provide a range of product add-ons from various vendors to support the entire range of communications requirements. VMS servers support DECnet, TCP/IP, and SNA as well as various asynchronous and serial communications protocols. MVS servers provide support for SNA, TCP/IP, and some support for other asynchronous communications.

Security at the server restricts access to software and data accessed from the server. Communications access is controlled from the communications server. In most implementations, the use of a user login ID is the primary means of security. Using LAN Server, some organizations have implemented integrated Response Access/Control Facility (RACF) security by creating profiles in the MVS environment and downloading those to the LAN server for domain control. Systems and network management services for the local LAN are managed by a LAN administrator, but WAN services must be provided from some central location. Typically, remote LAN management is done from the central data center site by trained MIS personnel.

4.3 Server Functionality in Detail

The discussion in the following sections more specifically describes the functions provided by the server in a NOS environment.

4.3.1 Request Processing

Requests are issued by a client to the NOS services software resident on the client machine. These services format the request into an appropriate RPC and issue the request to the application layer of the client protocol stack. This request is received by the application layer of the protocol stack on the server.

4.3.2 File Services

File services handle access to the virtual directories and files located on the client workstation and to the server's permanent storage. These services are provided through the redirection software implemented as part of the client workstation operating environment. As Chapter 3 described, all requests are mapped into the virtual pool of resources and redirected as necessary to the appropriate local or remote server. The file services provide this support at the remote server processor. In the typical implementation, software, shared data, databases, and backups are stored on disk, tape, and optical storage devices that are managed by the file server.

To minimize the effort and effect of installation and maintenance of software, software should be loaded from the server for execution on the client. New versions can be updated on the server and made immediately available to all users. In addition, installation in a central location reduces the effort required for each workstation user to handle the installation process. Because each client workstation user uses the same installation of the software, optional parameters are consistent, and remote help desk operators are aware of them. This simplifies the analysis that must occur to provide support. Sharing information, such as word processing documents, is easier when everyone is at the same release level and uses the same default setup within the software. Central productivity services such as style sheets and macros can be set up for general use. Most personal productivity products do permit local parameters such as eolors, default printers, and so forth to be set locally as well.

Backups of the server can be scheduled and monitored by a trained support person. Backups of client workstations can be scheduled from the server, and data can be stored at the server to facilitate recovery. Tape or optical backup units are typically used for backup; these devices can readily provide support for many users. Placing the server and its backups in a secure location helps prevent theft or accidental destruction of backups. A central location is readily monitored by a support person who ensures that the backup functions are completed. With more organizations looking at multimedia and image technology, large optical storage devices are most appropriately implemented as shared servers.

4.3.3 Fax/Print/Image Services

High-quality printers, workstation-generated faxes, and plotters are natural candidates for support from a shared server. The server can accept input from many clients, queue it according to the priority of the request and handle it when the device is available. Many organizations realize substantial savings by enabling users to generate fax output from their workstations and queue it at a fax server for transmission when the communication costs are lower. Incoming faxes can be queued at the server and transmitted to the appropriate client either on receipt or on request. In concert with workflow management techniques, images can be captured and distributed to the appropriate client workstation from the image server. In the client/server model, work queues are maintained at the server by a supervisor in concert with default algorithms that determine how to distribute the queued work.

Incoming paper mail can be converted to image form in the mail room and sent to the appropriate client through the LAN rather than through interoffice mail. Centralized capture and distribution enable images to be centrally indexed. This index can be maintained by the database services for all authorized users to query. In this way, images are captured once and are available for distribution immediately to all authorized users. Well-defined standards for electronic document management will allow this technology to become fully integrated into the desktop work environment. There are dramatic opportunities for cost savings and improvements in efficiency if this technology is properly implemented and used.

4.3.4 Database Services

Early database servers were actually file servers with a different interface. Products such as dBASE, Clipper, FoxPro, and Paradox execute the database engine primarily on the client machine and use the file services provided by the file server for record access and free space management. These are new and more powerful implementations of the original flat-file models with extracted indexes for direct record access. Currency control is managed by the application program, which issues lock requests and lock checks, and by the database server, which creates a lock table that is interrogated whenever a record access lock check is generated. Because access is at the record level, all records satisfying the primary key must be returned to the client workstation for filtering. There are no facilities to execute procedural code at the server, to execute joins, or to filter rows prior to returning them to the workstation. This lack of capability dramatically increases the likelihood of records being locked when several clients are accessing the same database and increases network traffic when many unnecessary rows are returned to the workstation only to be rejected.

The lack of server execution logic prevents these products from providing automatic partial update backout and recovery after an application, system, or hardware failure. For this reason, systems that operate in this environment require an experienced system support programmer to assist in the recovery after a failure. When the applications are very straightforward and require only a single row to be updated in each interaction, this recovery issue does not arise. However, many client/server applications are required to update more than a single row as part of one logical unit of work.

Client/server database engines such as Sybase, IBM's Database Manager, Ingres, Oracle, and Informix provide support at the server to execute SQL requests issued from the client workstation. The file services are still used for space allocation and basic directory services, but all other services are provided directly by the database server. Relational database management systems are the current technology for data management.

4.3.4.1 Flat Files: Sorting Physical Records

Database technology has evolved from the early 1960s' flat-file view when data was provided through punch cards or disk files simulating punch cards. These original implementations physically stored data columns and records according to the user view. The next column in the user view was the next column in the physical record, and the next record in the user view was the next physically stored record. Sorting the physical records provided the means by which a user was presented with a different view of related records. Columns were eliminated from view by copying the records from one location to another without the unnecessary columns. Many organizations today still use the flat-file approach to data management for reporting and batch update input. Data is extracted and sorted for efficient input to a batch report. Data is captured for update and sorted for more efficient input to a batch update program.

4.3.4.2 Hierarchical Databases: Adjacent Storage of Related Record Types

The second generation of database technology, the hierarchical database, could store related record types physically or logically next to each other. In the hierarchical model implementation, when a user accesses a physical record type, other application-related data is usually stored physically close and will be moved from disk to DRAM all together. Internally stored pointers are used to navigate from one record to the next if there is insufficient space close by at data creation time to insert the related data. Products such as IMS and IDMS implemented this technique very successfully in the early 1970s. Many organizations continue to use database applications built to use this technology.

The major disadvantage with the hierarchical technique is that only applications that access data according to its physical storage sequence benefit from locality of reference. Changes to application requirements that necessitate a different access approach require the data to be reorganized. This process, which involves reading, sorting, and rewriting the database into a new sequence, is not transparent to applications that rely on the original physical sequence. Indexes that provide direct access into the database provide the capability to view and access the information in a sequence other than the physical sequence. However, these indexes must be known to the user at the time the application is developed. The developer explicitly references the index to get to the data of interest. Thus, indexes cannot be added later without changing all programs that need this access to use the index directly. Indexes cannot be removed without changing programs that currently access the index. Most implementations force the application developer to be sensitive to the ordering and occurrence of columns within the record. Thus, columns cannot be added or removed without changing all programs that are sensitive to these records.

Application sensitivity to physical implementation is the main problem with hierarchical database systems. Application sensitivity to physical storage introduced considerable complexity into the navigation as application programmers traverse the hierarchy in search of their desired data. Attempts by database vendors to improve performance have usually increased the complexity of access. If life is too easy today, to create a bidirectionally virtually paired IMS logical relationship; that is why ôrganizations using products such as IMS and IDMS usually have highly paid database technical support staff

4.3.4.3 Relational Databases: Extracted Indexes and SQL

As hardware technology evolves, it is important for the data management capabilities to evolve to use the new capabilities. The relational database is the de facto standard today; therefore, investment by vendors will be in products that target and support fully compliant SQL databases.

Relational database technology provides the current data management solution to many of the problems inherent in the flat-file and hierarchical technologies. In the late 1970s and early 1980s, products such as Software AG's ADABAS and System 2000 were introduced in an attempt to provide the application flexibility demanded by the systems of the day. IBM with IMS and Cullinet with IDMS attempted to add features to their products to increase this flexibility. The first relational products were introduced by ADR with Datacom DB and Computer Corporation of America with Model 204.

Each of these implementations used extracted indexes to provide direct access to stored data without navigating the database or sorting flat files. All the products attempted to maintain some of the performance advantages afforded by locality of reference (storage of related columns and records as close as possible to the primary column and record).

Datacom and Model 204 introduced-for the first time=-the Structured Query Language (SQL). SQL was invented in the early 1970s by E. F. (Ted) Codd of IBM Labs in Santa Teresa, California. The primary design objective behind SQL was to provide a data access language that could be shown mathematically to manipulate the desired data correctly. The secondary objective was to remove any sense of the physical storage of data from the view of the user. SQL is another flat-file implementation; there are no embedded pointers. SQL uses extracted indexes to provide direct access to the rows (records) of the tables (files) of interest. Each column (field) may be used as part of the search criteria.

SQL provides (especially with SQL2 extensions) a very powerful data access language. Its algebra provides all the necessary syntax to define, secure, and access information in an SQL database. The elegance of the language intrigued the user and vendor community to the extent that standards committees picked up the language and defined a set of standards around the language syntax. SQLl and SQL2 define an exact syntax and a set of results for each operation. As a consequence, many software vendors have developed products that implement SQL. This standardization will eventually enable users to treat these products as commodities in the same way that PC hardware running DOS has become a commodity. Each engine will soon be capable of executing the same set of SQL requests and producing the same result. The products will then be differentiated based on their performance, cost, support, platform availability, and recovery-restart capabilities.

Dr. Codd has published a list of 13 rules that every SQL database engine should adhere to in order to be truly compliant. No products today can meet all of these criteria. The criteria, however, provide a useful objective set for the standards committees and vendors to strive for. We have defined another set of product standards that we are using to evaluate SQL database engines for the development of client/server applications, In particular, products should be implemented with support for the following products and standards:

ANSI SQL and IBM DB2 standards

A variety of front-end query products

C and COBOL SQL precompilers

Support for and compatibility with server NOS: NetWare, OS/2 (LAN Manager, LAN Server), Windows NT, Mac System 7, and/or UNIX (VINES, SCO, Sun, HP/UX USL, SVR4_), and MVS

Support for client Operating Systems: DOS, Windows, OS/2, Windows NT, Mac System 7, or UNIX (Solaris, USL, SCO, HP(UX, SVR4_)

Production-capable client/server database engines must be able to provide a similar operational environment to that found in the database engines present in minicomputer and mainframe computers today. Capabilities for comparison include performance, auditability, and recovery techniques. In particular, the foilowing DBMS features must be included in the database engine:

Performance optimization tools

Dynamic transaction backout

Roll back from, roll forward to last backup

Audit file recovery

Automatic error detection and recovery

File reclamation and repair tools

Support for mirrored databases

Capability to split database between physical disk drives

Remote distributed database management features

Maintenance of accurate and duplicate audit files on any LAN node

In the client/server implementation, you should offload database processing to the server. Therefore, the database engine should accept SQL requests from the client and execute them totally on the server, returning only the answer set to the client requestor. The database engine should provide support for stored procedures or triggers that run on the server.

The client/server model implies that there will be multiple concurrent user access: The database engine must be able to manage this access without requiring every developer to write well-behaved applications. The following features must be part of the database engine:

Locking mechanisms to guarantee data integrity

Deadlock detection and prevention

Multithreaded application processing

User access to multiple databases on multiple servers

4.3.4.4 Object-Oriented-A Brigbt Future

With the increasing maturity and popularity of OOTs for development, there has been a significant increase in maturity and acceptance of object-oriented database management systems (OODBMS). Object-oriented database management systems provide support for complex data structures: such as compound documents, CASE entity relationship models, financial models, and CAD/CAM drawings. OODBMS proponents claim that relational database management systems (RDBMS) can handle only simple data structures (such as tables) and simple transaction-processing applications that only need to create views combining a small number of tables. OODBMS proponents argue that there is a large class of problems that need to be and will be more simply implemented if more complex data structures can be viewed directly. RDBMS vendors agree with the need to support these data structures but argue that the issue is one of implementation, not architecture.

Relational databases are characterized by a simple data stmcture. All access to data and relationships between tables are based on values. A data value occurrence is uniquely determined by the concatenation of the table name, column name, and the value of the unique identifier of the row (the primary key). Relationships between tables are determined by a common occurrence of the primary key values. Applications build a view of information from tables by doing a join based on the common values. The result of the join is another table that contains a combination of column values from the tables involved in the join.

The development of a relational algebra defining the operations that can be performed between tables has enabled efficient implementations of RDBMSs. The establishment of industry standards for the definition of and access to relational tables has speeded the acceptance of RDBMSs as the de facto standard for all client/server applications today. Similar standards do not yet exist for OODBMSs. There is a place for both models. To be widely used, OODBMSs need to integrate transparently with RDBMS technology. Table 4.1 compares the terminology used by RDBMS and OODBMS proponents.

Table 4.1. Comparison of object-oriented and relational database management system features.

OODBMS	*RDBMS*
Class	Collection of rows in table
Object	Row
Type	Table definition (user type extension)
Method	Stored procedure (extension)
Index	Index
Object Identify	No match
Collection	Array (extension)
Inheritance	No match
Encapsulation	No match
Computationally complete	Transact SQL, PL/SQL, and stored procedures
No match	SQL portability
No match	Mathematically provable

There remain some applications for which RDBMSs have not achieved acceptable performance. Primarily, these are applications that require very complex data structures. Thousands of tables may be defined with many relationships among them. Frequently, the rows are sparsely populated, and the applications typically require many rows to be linked, often recursively, to produce the necessary view.

The major vendors in this market are Objectivity Inc., Object Design, Ontos, and Versant. Other vendors such as HP, Borland, and Ingres have incorporated object features into their products.

Only the logging functions will use real I/0. Periodically, O-RAM databases will be backed up to real magnetic or optical disk storage. During 1993, a significant number of production OODBMS applications were implemented. With the confidence and experience gained from these applications, the momentum is building, and 1994 and 1995 will see a significant increase in the use of OODBMSs for business critical applications. OODBMSs have reached a maturity level coincident with the demand for multimedia enabled applications. The complexities of dealing with multimedia demands the features of OODBMS for effective storage and manipulation.

To enable more complex data types to be manipulated by a single command, OODBMSs provide encapsulated processing logic with the object definition.

4.3.5 Communications Services

Client/server applications require LAN and WAN communication services. Basic LAN services are integral to the NOS. WAN services are provided by various communications server products. Chapter 5 provides a complete discussion of connectivity issues in the client/server model.

4.3.6 Security Services

Client/server applications require similar security services to those provided by host environments. Every user should be required to log in with a user ID and password. If passwords might become visible to unauthorized users, the security server should insist that passwords be changed regularly. The enterprise on the desk implies that a single logon ID and logon sequence is used to gain the authority once to access all information and process for the user has a need and right of access. Because data may be stored in a less physically secure area, the option should exist to store data in an encrypted form. A combination of the user ID and password should be required to decrypt the data.

New options, such as floppyless workstations with integrated data encryption standard (DES) coprocessors, are available from vendors such as Beaver Computer Company. These products automatically encrypt or decrypt data written or read to disk or a communication line. The encryption and decryption are done using the DES algorithm and the user password. This ensures that no unauthorized user can access stored data or communications data. This type of security is particularly useful for laptop computers participating in client/server applications, because laptops do not operate in surroundings with the same physical security of an office. To be able to access the system from a laptop without properly utilizing an ID number and password would be courting disaster.

4.4 The Network Operating System

The network operating system (NOS) provides the services not available from the client OS.

4.4.1 Novell NetWare

NetWare is a family of LAN products with support for IBM PC-compatible and Apple Macintosh clients, and IBM PC-compatible servers. NetWare is a proprietary NOS in the strict sense that it does not require another OS, such as DOS, Windows, Windows NT, OS/2, Mac System 7, or UNIX to run on a server. A separate Novell product-Portable NetWare for UNIX-provides server support for leading RISC-based UNIX implementations, IBM PC-compatible systems mnning Windows NT, OS/2, high-end Apple Macs running Mac System 7, and Digital Equipment Corporation VAXs running VMS.·

NetWare provides the premier LAN environment for file and printer resource sharing. It had 62 percent of the market share in 1993. It is widely installed as the standard product in many organizations. NetWare is the original LAN NOS for the PC world. As such, it incorporates many of the ease-of-use features required for sharing printers, data, software, and communications lines. Agreements between Novell and IBM to remarket the product and provide links between NetWare and the LAN Server product confirm the commitment to Novell Netware's use within large organizations.

Novell has committed to move NetWare to an open architecture. Through the use of open protocol technology (OPT), Novell makes NetWare fully network protocol independent. Two standardized interfaces-open datalink interface (OD!) and NetWare Streams=-enable other vendors to develop products for the NetWare environment. This facilitates its integration into other platforms.The diagram also illustrates the wide range of connectivity supported by NetWare. Client workstations can use Mac System 7, OS/2, DOS, Windows, Windows NT, NetWare, or UNIX NFS operating environments. OS/2, Windows NT, and UNIX servers may be installed on the same LAN as NetWare servers to provide support for products that require these platforms. Novell's purchase of USL from AT&T has increased its commitment to early support for native UNIX servers. HP, Sun, DEC, and Novell have announced an agreement to port NetWare to their respective UNIX platforms. Novell has won the battle to be *the* standard for the file/print server in the LAN environment.

Novell's published goal is to provide NetWare services totally independent of network media, network transport protocols, client/server protocols, and server and client operating systems, at each layer of network design.

NetWare has benefitted from its high performance and low resource requirements as much as it has from its relative ease of use. This performance has been provided through the use of a proprietary operating system and network protocols. Even though this has given Novell an advantage in performance, it has caused difficulties in the implementation of application and database servers in the Novell LAN. Standard applications cannot run on the server processor, because NetWare does not provide compatible APIs. Instead, NetWare provides a high performance capability called a NetWare Loadable Module (NLM) that enables database servers such as Sybase and Oracle, and communications servers such as Gateway Communications provides, to be linked into the NetWare NOS. In addition, the tailored operating environment does not provide some system features, such as storage protection and multitasking, in the same fundamental way that OS/2 and UNIX do. However, Novell is committed to address these issues by supporting the use of UNIX, OPENVMS, OS/2, and Windows NT as native operating environments.

With the release ofNetW are 4.0, Novell addressed the serious issue of enterprise computing with improved network directory services (NDS), one thousand node domains, and LAN/WAN support for TCP/IP. Native NetWare 4.x will be available to developers in early 1994 and production ready by the end of 1994. For the other end of the product range, Novell released NetWare Lite in 1993 to address the small business and simple workgroup requirements of LANs with five or fewer workstations. This enables organizations to remain with NetWare as the single LAN technology everywhere. Clearly, Novell's pitch is that systems management and administration are greatly simplified with the single standard of "NetWare Everywhere."

4.4.2 LAN Manager

LAN Manager and its IBM derivative, LAN Server, are the standard products for use in client/server implementations using OS/2 as the server operating system. LAN Manager/X is the standard product for client/server implementations using UNIX System V as the server operating system. Microsoft released its Advanced Server product with Windows NT in the third quarter of 1993. During 1994, it will be enhanced with support for the Microsoft network management services, currently referred to as "Hermes," and Banyan's Enterprise Network Services (ENS). Advanced

Server is the natural migration path for existing Microsoft LAN Manager and IBM LAN Server customers. Existing LAN Manager/X customers probably won't find Advanced Server an answer to their dreams before 1995.

AT&T has taken over responsibility for the LAN Manager/X version. Vendors such as Hewlett-Packard (HP) have relicensed the product from AT&T. AT&T and Microsoft have an agreement to maintain compatible APis for all base functionality.

LAN Manager and Advanced Server provide client support for DOS, Windows, Windows NT, OS/2, and Mac System 7. Server support extends to NetWare, AppleTalk, UNIX, Windows NT, and OS/2. Client workstations can access data from both NetWare and LAN Manager servers at the same time. LAN Manager supports NetBIOS and Named Pipes LAN communications between clients and OS/2 servers. Redirection services are provided to map files and printers from remote workstations for client use.

Advanced Server also supports TCP/IP communication. In early 1994, Advanced Server still will be a young product with many missing pieces. Even more troublesome, competitiveness between Microsoft and Novell is delaying the release of client requestor software and NetWare Core Protocol (NCP) support. Microsoft has added TCP/IP support to LAN Manager 2.1 and Advanced Server along with NetView and Simple Network Management Protocol (SNMP) agents. Thus, the tools are in place to provide remote LAN management for LAN Manager LANs. Microsoft has announced support for IBM NetView 6000 for Advanced Server management.

Advanced Server provides integrated support for peer-to-peer processing and client/server applications. Existing support for Windows NT, OS/2, UNIX, and Mac System 7 clients lets application, database, and communication servers run on the same machine as the file and print server. This feature is attractive in small LANs. The native operating system support for preemptive multitasking and storage protection ensures that these server applications do not reduce the reliability of other services. Even as Windows NT is rolled out to provide the database, application, and communications services to client/server applications, the use of Novell as the LAN NOS of choice will continue for peripheral resource sharing applications.

Microsoft has attempted to preempt the small LAN market with its Windows for Workgroups (WfW) product. This attacks the same market as NetWare Lite with a low-cost product that is tightly integrated with Windows. It is an attractive option for small organizations without a requirement for larger LANs. The complexities of systems management make it less attractive in an enterprise environment already using Novell. WfW can be used in conjunction with Novell for a workgroup wishing to use some WfW services, such as group scheduling.

4.4.3 IBM LAN Server

IBM has entered into an agreement to resell and integrate the Novell NetWare product into environments where both IBM LAN Server and Novell NetWare are required. NetWare provides more functional, easier-to-use, and higher-performance file and print services. In environments where these are the only LAN functions, NetWare is preferable to LAN Manager derivatives. The capability to interconnect to the SNA world makes the IBM product LAN Server attractive to organizations that prefer to run both products. Most large organizations have department workgroups that require only the services that Novell provides well but may use LAN Server for client/server applications using SNA services such as APPN.

IBM and Microsoft had an agreement to make the APis for the two products equivalent. However, the dispute between the two companies over Windows 3.x and OS/2 has ended this cooperation. The most recent releases of LAN Manager NT 3 and LAN Server 3 are closer to the agreed equivalency, but there is no guarantee that this will continue. In fact, there is every indication that the products will diverge with the differing server operating system focuses for the two companies. IBM has priced LAN Server very attractively so that if OS/2 clients are being used, LAN Server is a low-cost option for small LANs. LAN Server supports DOS, Windows, and OS/2 clients. No support has been announced for Mac System 7, although it is possible to interconnect AppleTalk and LAN Server LANs to share data files and communication services.

4.4.4 Banyan VINES

Banyan VINES provides basic file and print services similar to those of Novell and Lan Manager. VINES incorporates a facility called Street'I'alk that enables every resource in a Banyan enterprise LAN to be addressed by name. VINES also provides intelligent WAN routing within the communications server component. These two features are similar to the OSI Directory Services X.500 protocol. StreetTalk enables resources to be uniquely identified on the network, making them easier to access and manage. All resources, including file services, users, and printers, are defined as objects. Each object has a StreetTalk name associated with it.

StreetTalk names follow a three-level hierarchical format: Item@Grnup@Organization. For example, a user can be identified as

Psmith@Cerritos@Tnet. All network objects are stored in a distributed database that can be accessed globally. Novell's NDS is similar to Street'I'alk in functionality. However, there are key differences. NDS can partition and replicate the database, which will generally improve performance and reliability. NDS is X.500-compliant and enables multiple levels of hierarchy.

StreetTalk supports a fixed three-level hierarchy. TheNDS architecture offers more flexibility but with corresponding complexity, and StreetTalk is less flexible but less complex to manage. One advantage the current version of StreetTalk has over NDS is that StreetTalk objects can have unlimited attributes available for selection. To locate a printer with certain attributes, the command: "Locate a color laser printer with A4 forms on the 7th floor of Cerritos" finds and uses the printer with the desired characteristics.

VINES V5.5 offers ISDN and TI support for server-to-server communications over a WAN, as well as integration of DOS, Windows, OS/2, and Mac clients. VINES does not support NFS clients. Novell and Microsoft have announced support for Banyan ENS within their products to be available in Q2 1994. Banyan and DCA provide SNA services to the VINES environment. VINES supports UNIX, DOS, Windows, OS/2, and Mac System 7 clients.

4.4.5 PC Network File Services (NFS)

NFS is the standard file system support for UNIX. PC NFS is available from SunSelect and FTP to provide file services support from a UNIX server to Windows, OS/2, Mac, and UNIX clients.

NFS lets a client mount an NFS host's filing system (or a part of it) as an extension of its own resources. NFS's resource-sharing mechanisms encompass interhost printing. The transactions among NFS systems traditionally ride across TCP/IP and Ethernet, but NFS works with any network that supports 802.3 frames.

SunSelect includes instructions for adding PC-NFS to an existing LAN Manager or Windows for Workgroups network using Network Driver Interface Specification (NDIJS) drivers.With the increasing use of UNIX servers for application and database services, there is an increasing realization that PC NFS may be all that is required for NOS support for many workgroups. This can be a low-cost and low-maintenance option because the UNIX server is easily visible from a remote location.

4.5 What Are the Available Platforms?

Client/server computing requires that LAN and WAN topologies be in place to provide the necessary intemetworking for shared applications and data. Gartner Group 1 surveyed and estimated the Microsystems' integration topologies for the period 1986-1996; the results appear in Figure 4.6. Of special interest is the projection that most workstations will be within LANs by 1996, but only 14 percent-will be involved in an enterprise LAN by that date. These figures represent a fairly pessimistic outlook for interconnected LAN-to-LAN and enterprise-wide connectivity. These figures probably will prove to be substantially understated if organizations adopt an architectural perspective for the selection of their platforms and tools and use these tools within an organizationally optimized systems development environment (SDE).

4.5.1 Workstations in LAN Configuration

This model is the most basic implementation providing the standard LAN services for file and printer sharing.

4.5.2 LAN-to-LAN/WAN Configuration

Routers and communication servers will be used to provide communication services between LANs and into the WAN. In the client/server model, these eonnections will be provided transparently by the SDE tools. There are significant performance implications if the traffic volumes are large. IBM's LU6.2 implementation in APPC and TCP/IP provides the best support for high-volume, LAN-to-LAN/WAN communications. DEC's implementation of DECnet always has provided excellent LAN-to-WAN connectivity. Integrated support for TCP/IP, LU6.2, and IPX provides a solid platform for client/server LAN-to-WAN implementation within DECnet. Novell 4.x provides support for TCP/IP as both the LAN and WAN protocol. Internetworking also is supported between IPX and TCP/IP.

4.5.3 LAN-to-Host Configuration

The lack of real estate on the desktop encouraged most organizations to move to a single device-using terminal emulation from the workstation-to access existing mainframe applications. It will take considerable time and effort before all existing host-based applications in an organization are replaced by client/server applications. In the long term, the host will continue to be the location of choice for enterprise database storage and for the provision of security and network management services.

Mainframes are expensive to buy and maintain, hard to use, inflexible, and large, but they provide the stability and capacity required by many organizations to run their businesses. Their roles will change, but they will be around as part of the enterprise in:frastmcture for many more years. Only organizations who create an enterprise architecture strategy and transformational plans will accomplish the migration to client/server in less than a few years. Without a well-architected strategy, gradual evolution will produce failure.

4.5.4 Enterprise-Wide

Information that is of value or interest to the entire business must be managed by a central data administration function and appear to be stored on each user's desk. These applications are traditionally implemented as Online Transaction Processing (OLTP) to the mainframe or minicomputer. With the client/server model, it is feasible to use database technology to replicate or migrate data to distributed servers. Wherever data resides or is used, the location must be transparent to the user and the developer. Data should be stored where it best meets the business need.

4.5.4.lOLTP on a LAN

Online Transaction Processing applications are found in such industries as insuranee, finance, government, and sales-all of which process large numbers of transactions. Each of these transactions requires a minimal amount of user think time to process. In these industries, data is frequently collected at the source by the knowledgeable worker. As such, the systems have high requirements for availability, data integrity, performance, concurrent access, growth potential, security, and manageability. Systems implemented in these environments must prove their worth or they will be rejected by an empowered organization. They must be implemented as an integral part of the job process.

OLTP has traditionally been the domain of the large mainframe vendors-such as IBM and DEC-and of special-purpose, fault-tolerant processors from vendors such as Tandem and Stratus. The client/server model has the capability to provide all the services required for OLTP at much lower cost than the traditional platforms. All the standard client/server requirements for a GUI-application portability, client/server function partitioning, software distribution, and effective development tools-exist for OLTP applications.

The first vendor to deliver a production-quality product in this arena is Cooperative Solutions with its Ellipse product. Prior to Ellipse, OLTP systems required developers to manage the integrity issues of unit-of-work processing, including currency control and transaction rollback. Ellipse provides all the necessary components to build systems with these features. Ellipse currently operates with Windows 3.x, OS/2 clients, and OS/2 servers using the Sybase database engine. Novell is working with Cooperative Solutions to port Ellipse as a Novell NetWare Loadable Module (NLM). It provides a powerful GUI development environment using a template language as a shorthand for development. This language provides a solid basis for building an organizational SDE and lends itself well to the.incorporation of standard components.

4.5.4.2 OLTP with. UNIX

As UNIX has matured, it has added many of the features found in other commercial operating systems such as VMS and MVS. There are now several offerings for OLTP with UNIX. IBM is promoting CICS 6000 as a downsizing strategy for CICS MVS. Database services will be provided by a combination of AIX and MYS servers.

Novell purchased the Tuxedo product from AT&T with its acquisition of USL. OSF selected the Transarc Ensina product as the basis for OLTP with DCE. The DCE recognition quickly placed Ensina in the lead in terms of supported UNIX platforms. IB,M has released a version of DCE for AIX that includes the Ensina technology. NCR provides a product called TopEnd as part of its Cooperation series.

Client/server TP monitor software is becoming increasingly necessary now that client/server systems are growing to include several database servers supporting differentvendors' databases and servicing tens, hundreds, and even thousands of users that need to access and update the same data. UNIX-based OTLP products are maturing to provide the same level of functionality and reliability as traditional mainframe-based IBM Customer Information Control Systems (CICS), yet at less cost and with graphical front ends.

4.6 The Server Operating System

Servers provide the platform for application, database, and communication services. There are six operating system platforms that have the greatest potentional and/or are prevalent today: NetWare, OS/2, Windows NT, MVS, VMS, and UNIX

4.6.1 NetWare

NetWare is used by many organizations, large and small, for the provision of file, printer, and network services. NetWare is a self-contained operating system. It does not require a separate OS (as do Windows NT, OS/2, and UNIX) to run. Novell is taking steps to allow NetWare to run on servers with UNIX. Novell purchased USL and will develop shrink-wrapped products to run under both NetWare and UNIX System V, Release 4.2. The products will enable UNIX to simultaneously access information from both a NetWare and a UNIX server.

4.6.2 OS/2

OS/2 is the server platform for Intel products provided by IBM in the System Application Architecture (SAA) model. OS/2 provides the storage protection and preemptive multitasking services needed fo/tiie server platform. Several database and many application products have been ported to OS/2. The only network operating systems directly supported with OS/2 are LAN Manager and LAN Server. Novell supports the use of OS/2 servers running on separate processors from the NetWare server. The combination of Novell with an OS/2 database and application servers can provide the necessary environment for a production-quality client/server implementation. Appendix A describes such an implementation.

4.6.3 Windows NT

With the release of Windows NT (New Technology) in September of 1993, Microsoft staked its unique position with a server operating system. Microsoft's previous development of OS/2 with IBM did not create the single standard UNIX ,alternative that was hoped for. NT provides the preemptive multitasking services required for a functional server. It provides excellent support for Windows clients and incorporates the necessary storage protection services required for a reliable server operating system. Its implementation of C2 level security goes well beyond that provided by OS/2 and most UNIX implementations. It will take most of 1994 to get the applications and ruggedizing necessary to provide an industrial strength platform for business critical applications. With Microsoft's prestige and marketing muscle, NT will be installed by many organizations as their server of choice.

4.6.4MVS

IBM provides MVS as a platform for large applications. Many of the existing application services that organizations have purchased operate on System 370-compatible hardware running MVS. The standard networking environment for many large organizations-SN A-is a component of MVS. IBM prefers to label proprietary systems today under the umbrella of SAA. The objective of SAA is to provide all services on all IBM platforms in a compatible way-the IBM version of the single-system image.

There is a commitment by IBM to provide support for the LAN Server running natively under MVS. This is an attractive option for organizations with large existing investments in MVS applications. The very large data storage capabilities provided by System 370-compatible platforms with MVS make the use of MVS for LAN services attractive to large organizations. MYS provides a powerful database server using DB2 and LU6.2. With broad industry support for LU6.2, requests that include DB2 databases as part of their view can be issued from a client/server application. Products such as Sybase provide high-performance static SQL support, making this implementation viable for high-performance production applications.

4.6.5 OPENVMS

Digital Equipment Corporation provides OPENVMS as its server platform of choice. VMS has a long history in the distributed computing arena and includes many of the features necessary to act as a server in the client/server model. DEC was slow to realize the importance of this technology, and only recently did the company enter the arena as a serious vendor. NetWare supports the use of OPENVMS servers for file services. DEC provides its own server interface using a LAN Manager derivative product called Pathworks.

Pathworks runs native on the VAX and RISC Alpha RXP. This is a particularly attractive configuration because it provides access on the same processor to the application, database, and file services provided by a combination of OPENVMS, NetWare, and LAN Manager. Digital and Microsoft have announced joint agreements to work together to provide a smooth integration of Windows, Windows NT, Pathworks, and OPENVMS. This will greatly facilitate the migration by OPENVMS customers to the client/server model.

VAX OPENVMS support for database products such as RDB, Sybase, Ingres, and Oracle enables this platform to execute effectively as a database server for client/server applications. Many organizations have large investments in VAX hardware and DECnet networking. The option to use these as part of client/server applications is attractive as a way to maximize the value of this investment. DECnet provides ideal support for the single-system image model. LAN technology is fundamental to the architecture of DECnet. Many large organizations moving into the client/server world of computing have standardized on DECnet for WAN processing. For example, Kodak selected Digital as its networking company even after selecting IBM as its mainframe outsourcing company.

4.6.6 UNIX

UNIX is a primary player as a server system in the client/server model. Certainly, the history of UNIX in the distributed computing arena and its open interfaces provide an excellent opportunity for it to be a server of choice. To understand what makes it an open operating system, look at the system's components. UNIX was conceived in the early 1970s by AT&T employees as an operating environment to provide services to software developers who were discouraged by the incompatibility of new computers and the lack of development tools for application development. The original intention of the UNIX architecture was to define a standard set of services to be provided by the UNIX kernel. These services are used by a shell that provides the command-line interface. Functionality is enhanced through the provision of a library of programs. Applications are built up from the program library and custom code. The power and appeal of UNIX lie in the common definition of the kernel and shell and in the large amount of software that has been built and is available. Applications built around these standards can be ported to many different hardware platforms.

The objectives of the original UNIX were very comprehensive and might have been achieved except that the original operating system was developed under the auspices of AT&T. Legal ramifications of the consent decree governing the breakup of the Regional Bell Operating Companies (RBOCs) prevented AT&T from getting into the computer business. As a result, the company had little motivation early on to promote UNIX as a product.

To overcome this, and in an attempt to achieve an implementation of UNIX better suited to the needs of developers, the University of California at Berkeley and other institutions developed better varieties of UNIX. As a result, the original objective of a portable platform was compromised. The new products were surely better, but they were not compatible with each other or the original implementation. Through the mid-1980s, many versions of UNIX that had increasing functionality were released. IBM, of course, entered the fray in 1986 with its own UNIX derivative, AlX. Finally, in 1989, an agreement was reached on the basic UNIX kernel, shell functions, and APis.

The computing community is close to consensus on what the UNIX kernel and shell will look like and on the definition of the specific APis. During all of these gyrations, one major UNIX problem has persisted that differentiates it from DOS, Windows NT, and OS/2 in the client/server world. Because the hardware platforms on which UNIX resides come from many manufacturers and are based on many different chip sets, the "off-the-shelf" software that is sold for PCs is not yet available for UNIX Software is sold and distributed in its executable form, so it must be compiled and linked by the developer for the target platform. This means that organizations wishing to buy UNIX software must buy it for the specific target platform they are using. This also means that<\~hen they use many platforms in a distributed client/server application, companies must buy different software versions for each platform.

In addition to the complexity this entails, a more serious problem exists with software versioning. Software vendors update their software on a regular basis, adding functionality and fixing problems. Because the UNIX kernel is implemented on each .platfoım and the software must be compiled for the target platform, there are differences in the low-level operation of each platform. This requires that software vendors port their applications to each platform they suppoıt. This porting function can take from several days to several months. In fact, if the platform is no longer popular, the port may never occur. Thus, users who acquire a UNIX processor may find that their software vendor is no longer committed to upgrading their software for this platform.

The major UNIX developer groups-UNIX International, Open Systems Foundation (OSF), and X/Open-have worked on plans to develop a binary compatible UNIX. If and when this happens, every new processor will execute the same metamaohine language. Despite the fact that at the machine level there will be differences, the executable code will be in this metalanguage. Software developers then will be able to develop off-the-shelf UNIX applications. When we achieve this level of compatibility, the true promise of UNIX will be reached, and its popularity should take off. The Open Software Foundation (OSF), a nonprofit consortium founded in 1988, now encompasses 74 companies, including Computer Associates, DEC, Groupe Bull, HP, IBM, Microsoft, Novell, Nippon Telegraph and Telephone Corp., Siemens Nixdorf, and even UNIX International Inc. (which was the standards-setting group for AT&T's, then X/Open's, UNLX System V). The OSF has set a goal to build distributed computing environment (DCE) compatibility into its distributed computing architecture. The OSF aims to provide an X/Open and POSIX compliant UNIX-like operating system using the Motif graphical user interface. The OSF has developed the Architecture Neutral Distribution Format (ANDF) with the intention of providing the capability to create and distribute shrink-wrapped software that can run on a variety of vendor platforms. The first operating system version OSF/1 was delivered by OSF in 1992 and implemented by DEC in 1993.

The important technologies defined for OSF include

Remote procedure call (RPC) services

Data-sharing services

E:--mail naming

Security software

Time services

Parallel programming support

Data-sharing and print services

DOS file and printer clients

UNIX is particularly desirable as a server platform for client/server computing because of the large range of platform sizes available and the huge base of application and development software available. Universities are contributing to the UNIX momentum by graduating students who see only UNIX during their student years. Government agencies are insisting on UNIX as the platform for all government projects. The combination of these pressures and technology changes should ensure that UNIX compatibility will be mandatory for server platforms in the last half of this decade.

OSF initially developed Motif, a graphical user interface for UNIX, that has become the de facto UNIX GUI standard. The Distributed Computing Environment (DCE) is gaining acceptance as the standard for distributed application development although its Distributed Management Environment has yet to achieve such widespread support, OSF/1, the OSF defined UNIX kernel, has been adopted only by DEC, although most other vendors have made promises to support it. OSF/1 brings the promise of a UNIX micro kernel more suitable to the desktop environment than existing products.

The desire for a standard UNIX encourages other organizations. For example, the IEEE tackled the unified UNIX issue by establishing a group to develop a standard portable operating system called POSIX. The objective is to develop an ANSI standard operating system. POSIX isn't UNIX, but it is UNIX-like. POSIX standards (to which most vendors pledge compliance) exist today. DEC's OPENVMS operating system, for exampie, supports published POSIX standards. POSIX at this point, however, does little to promote interoperability and portability because so little of the total standard has been finalized. Simple applications that will run across different POSIX-compliant platforms be written. However, they will be limited applications because developers will be unable to use any of the rich, non-POSIX features and functions that the vendors offer beyond the basic POSIX-compliant core.

X/Open started in Europe and has spread to include most major U.S. computer makers. X/Open is having significant impact in the market because its goal is to establish a standard set of Application Programming Interfaces (APis) that will enable interoperability. These interfaces are published in the X/Open Portability Guide. Applications running on operating systems that comply with these interfaces will communicate with each other and interoperate, even if the underlying operating systems are different. This is the key objective of the client/server model.

The COSE announcement by HP, IBM, SCO, Sun, and Univel (Novell/USL) in March 1993 at the Uniforum Conference is the latest attempt to create a common ground between UNIX operating systems. The initial COSE announcement addresses only the user's desktop environment and graphical user interface, although in time it is expected to go further. COSE is a more pragmatic group attempting to actually "get it done." Another major difference from previous attempts to create universal UNIX standards is the involvement of SCO and Sun. These two organizations own a substantial share of the UNIX market and have tended to promote proprietary approaches to the desktop interface. SCO provides its Open Desktop environment, and Sun offers Open Look. The commitment to Motif is a significant concession on their part and offers the first real opportunity for complete vendor interoperability and user transparency to platform.

In October of 1993, Novell agreed to give the rights to the UNIX name to X/Open so that all vendors can develop to the UNIX standards and use the UNIX name for their products. This largely symbolic gesture will eliminate some of the confusion in the marketplace over what software is really UNIX. COSE is looking beyond the desktop to graphics, multimedia, object technology, and systems management. Networking support includes Novell's NetWare UNIX client networking products, OSF's DCE, and SunSoft's Open Network Computing. Novell has agreed to submit the NetWare UNIX client to X/Open for publication as a standard. In the area of graphics, COSE participants plan to support a core set of graphics facilities from the X Consortium, the developer of X Windows.

Addressing multimedia, the COSE participants plan to submit two joint specifications in response to the Interactive Multimedia Association's request for technology. One of those specifications, called Distributed Media Services (DMS), defines a network-independent infrastructure supporting an integrated API and data stream protocol. The other-the Desktop Integrated Media Environment-will define multimedia access and collaboration tools, including at least one basic tool for each data type: supported by the DMS infrastructure. The resulting standard will provide users with consistent access to multimedia tools in multivendor environments.

COSE also addresses object technology, an area targeted by IBM and Sun. The group will support the efforts of the Object Management Group (OMG) and its Common Object Request Broker (CORBA) standard for deploying and using distributed objects. IBM already has a COREA-compliant object system in beta test for A!X. Sun built an operating system code named Spring as a proof of concept in 1992. Sun has a major project underway, called Distributed Objects Everywhere (DOE), that is producing very exciting productivity results. Finally, COSE will focus on the management of distributed file systems, distribution, groups and users, print spooling, software installation licensing, and storage.

It is not a coincidence that these vendors are coming together to define a standard UNIX at this time. The COSE effort is a defensive reaction to the release of Microsoft's Windows NT. With this commitment to a 32-bit desktop and server eperating system, Microsoft has taken the wind out of many of the UNIX claims to 'technical superiority. Despite its numerous advantages as a desktop and server operating system, UNIX never has been widely accepted in the general corporate world that favors DOS/Windows and Novell's Net Ware. A key drawback to UNIX in the corporate arena has been the lack of a single UNIX standard. UNIX has a well established position as the operating system of choice for distributed relational databases from vendors like Informix, Ingres, Oracle, and Sybase. Most of these vendors, however, will port their products to Windows NT as well. Any effort to reduce the problems associated with the multiple UNIX variants will do much to bolster the stature of UNIX as a worthwhile alternative to Windows NT.

4.6.7 Distributed Computing Environment (DCE)

Spin this fantasy around in your mind. All the major hardware and software vendors get together and agree to install a black box in their systems that will, in effect, wipe away their technological barriers. This black box will connect a variety of small operating systems, dissimilar hardware platforms, incompatible communications protocols, all sorts of applications and database systems, and even unlike security systems. And the black box will do all this transparently, not only for end users but also for systems managers and applications developers.2 OSF proposes the distributed computing environment (DCB) as this black box. DCB is the most important architecture defined for the client/server model. It provides the bridge between existing investments in applications and new applications based on current technology.

The first product components of DCB were released in the third quarter of 1991. DCB competes directly with Sun's open network computing (ONC) environment and indirectly with many other network standards. OSF/1 and DCE are almost certain to win this battle because of the massive market presence of the OSF sponsors. IBM has now committed to making its AIX product OSF/1 compatible by early 1994. It will be 1995 bef.. the product is mature and complete enough to be widely used as part of business applications. In the interim, product vendors and systems integrators will use it to build portable products and applications. The general availability of code developed for previous, similar product components will speed the process and enable new development to be modelled on the previous releases.

DCE is a prepackaged group of integrated interoperability applications that connect diverse hardware and software systems, applications, and databases. To provide these services, DCE components must be present on every platform in a system. These components become active whenever a local application requests data, services, or processes from somewhere. The OSF says that DCE will make a network of systems from multiple vendors appear as a single stand-alone computer to applications developers, systems administrators, and end users. Thus, the single-system image is attained.

The various elements of DCB are as follows:

Remote Procedure Call (RPC) and Presentation Services: Interface Definition Languages (IDLs) and RPCs enable programmers to transfer control and data across a network in a transparent manner that helps to mask the network's complexity. DCE uses the RPC originally developed by the HP Apollo Network Computing System (NCS), with some enhancements by DEC and IBM. NCS also provides the Network Data Representation (NDR), a virtual data representation. NDR enables data to be exchanged between various vendor products transparently. Conversions (as necessary) will take place with no intervention by the caller. Naming, security, file system, and data type conversions may take place as data is transported between various platforms.

Naming: User-oriented names, specifying computers, files, and people should be easily accessible in a distributed environment. These directory services must offer standard appearance and rules for all clients. DCB supports the X 500 directory services standard, adding extensions from DEC's Domain Name Service (DECdns). The standardized X.500 code is Siemens Nixdorf's DlR-X X.500 service.

Security: Distributed applications and services must identify users, control access to resources, and guard the integrity of all applications. DCE uses the Kerberos authentication service, developed by MIT as part of its Athena networking project and enhanced by Hewlett-Packard. This service is one of the major challenges to making products available quickly, because very few products today are developed with an awareness of this specification.

Threads: This terminology represents a method of supporting parallel execution by managing multiple threads of control within a process operating in a distributed environment. Threads enable systems to start up multiple processes and forget about them until they are completed. This is especially important for network servers that may have to handle many requests from many clients at the same time. They must be able to do this without waiting for the previous request to complete. DCB is using DEC's Concert Multithread Architecture (CMA) implementation.

Time Service: A time service synchronizes all system clocks of a distributed environment so that executing applications can depend on equivalent clocking among processes. Consider that many machines operating in many time zones may provide processes as part of a single application solution. It is essential that they agree on the time in order to manage scheduled events and time-sequenced events. DCB is using a modification of DEC's Distributed Time Synchronization Service (DECdts).

Distributed File Services: By extending the local file system throughout the network, users gain full access to files on remote configurations. DCB uses Sun's Network File System (NFS) Version 2 and provides next-generation capabilities with the Andrew File System (AFS), developed at Carnegie-Mellon University and commercialized by Transarc Corp. Diskless operations under AFS are supported by development work done by Hewlett-Packard.

PC Integration: Integration enables PCs using MS-DOS, Windows NT, and OS/2 to access file and print services outside the MS-DOS environment. DCE uses Microsoft's LAN Manager/X.

Management: Although partly addressed by the previous components, management is so complex in a distributed, heterogeneous configuration that OSF has defined a new architecture: distributed management environment (DME). DME provides a common framework for the management of stand-alone and distributed systems. This framework provides consistent tools and techniques for managing different types of systems and enables vendors to build system management applications that work on a variety of platforms. OSF will base DME on technology from Hewlett-Packard's OpenView product.

Communications: DCB is committed to support the OSI protocol stack.

4.7 System Application Architecture (SAA)

SAA is IBM's distributed environment. SAA was defined by IBM in 1986 as an architecture to integrate all IBM computers and operating systems, including MVS, VM/CMS, OS/400, and OS/2-EE. SAA defines standards for a common user access (CUA) method, common programming interfaces (CPI), and a common communication link (APPC).To support the development of SAA-compliant applications, IBM described SAA frameworks (that somewhat resemble APis). The first SAA framework AD/Cycle, the SAA strategy for CASE application development. AD/Cycle is designed to use third-party tools within the IBM SAA hardware and mainframe Repository Manager/MYS data storage facility. Several vendors have been selected by IBM as AD/Cycle partners, namely: Intersolv, KnowledgeWare, Bachman, Synon, Systematica,and Easel Corp. Several products are already available, including the Easel WorkBench toolkit, Bachman DB2, CSP tools, and the KnowledgeWare Repository and MVStools.

Unfortunately, the most important component, the Repository Manager, has not yet reached production quality in its MVS implementation and as yet there are no plans for a client/server implementation. Many original IBM customers involved in evaluating the Repository Manager have returned the product in frnstration. Recently, there has been much discussion about the need for a production-quality, object-oriented database management system (OODBMS) to support the entity relationship (ER) model underlying the repository. Only this, say some sources, will make implementation and performance practical. A further failing in the SAA strategy is the lack of open systems support. Although certain standards, such as Motif, SQL, and LU6.2, are identified as part of SAA; the lack of support for AIX has prevented many organizations from adopting SAA. IBM has published all the SAA standards and has licensed various protocols, such as LU6.2. The company has attempted to open up the SAA software development world. IBM's director of open systems strategy, George Siegle, says that IBM believes in openness through interfaces. Thus, the complete definition of APis enables other vendors to develop products that interface with IBM products and with each other. Recent announcements, such as support for CICS AIX, point to a gradual movement to include AIX in the SAA platforms. The first SAA application that IBM released, Office'Vision, was a disaster. The product consistently missed shipping dates and lacked much of the promised functionality. IBM has largely abandoned the product now and is working closely with Lotus and its workgroup computing initiatives.

IBM has consistently defined common database, user interface, and communications standards across all platforms. This certainly provides the opportunity to build SAA-compliant client/server applications. The recent introduction of CICS for OS/2, AIX, and OS/400 and the announcement of support for AIX mean that a single transaction-processing platform is defined across the entire range of products. Applications developed under OS/2 can be ported to interoperate between OS/2, OS/400, MVS, and eventually AIX, without modification. COBOL and Care common programming languages for each platform. SQL is the common data access language in all platforms.

The failure of SAA is attributable to the complexity of IBM's heterogenous product lines and the desire of many organizations to move away from proprietary to open systems solutions. This recognition led IBM to announce its new Open Enterprise plan to replace the old System Application Architecture (SAA) plan with an open network strategy. SystemView is a key IBM network product linking OS/2, UNIX, and AS/400 operating systems. Traditional Systems Network Architecture (SNA) networking will be replaced by new technologies, such as Advanced Peer-to-Peer Communications (APPC) and Advanced Peer-to-Peer Networking (APPN).

IBM has defined SystemView as its DME product. SystemView defines APIs to enable interoperability between various vendor products. It is expected to be the vehicle for linking AIX into centralized mainframe sites. IBM has stated that SystemView is an open structure for integrating OSI, SNA, and TCP/IP networks. At this time, SystemView is a set of guidelines to help third-party software developers and customers integrate systems and storage management applications, data definitions, and access methods. The guidelines are intended to further support single-system image concepts.

---5---

Components of Client/Server Applications

---Connectivity---

5.1 Executive Summary

The network is the computer is the most appropriate description of client/server computing. Users want to feel that somewhere on the network the services they need are available and are accessible based on a need and right of access, without regard to the technologies involved. When ready to move beyond personal productivity stand-alone applications and into client/server applications, organ-izations must address the issues of connectivity. Initially, most users discover their need to access a printer that is not physically connected to their client workstation. Sharing data files among non-networked individuals in the same office can be handled by "sneakemet" (hand-carrying diskettes), but printing is more awkward. The first LANs installed are usually basic networking services to support this printer-sharing requirement. Now a printer anywhere in the local area can be authorized for shared use.

The physical medium to accomplish this connection is the LAN cabling. Each workstation is connected to a cable that routes the transmission either directly to the next workstation on the LAN or to a hub point that routes the transmission to the appropriate destination. There are two primary LAN topologies that use Ethernet (bus) and Token Ring (ring).

Ethernet and Token Ring are implemented on well-defined Institute of Electrical and Electronic Engineers (IEEE) industry standards. These standards define the product specification detail and provide a commitment to a fixed specification. This standardization has encouraged hundreds of vendors to develop competitive products and in tum has caused the functionality, performance, and cost of these LAN connectivity products to improve dramatically over the last five years. Older LAN installations that use nonstandard topologies (such as ARCnet) will eventually require replacement.

There is a basic functional difference in the way Ethernet and Token Ring topologies place data on the cable. With the Ethernet protocol, the processor attempts to dump data onto the cable whenever it requires service. Workstations contend for the bandwidth with these attempts, and the Ethernet protocol includes the appropriate logic to resolve collisions when they occur. On the other hand, with the Token Ring protocol, the processor only attempts to put data onto the cable when there is capacity on the cable to accept the transmission. Workstations pass along a *token* that sequentially gives each workstation the right to put data on the network.

Recent enhancements in the capabilities of intelligent hubs have changed the way we design LANs. Hubs owe their success to the efficiency and robustness of the 10BaseT protocol, which enables the implementation of Ethernet in a star fashion over Unshielded Twisted Pair (UTP) wiring. Now commonly used, hubs provide integrated support for the different standard topologies such as Ethernet, Token Ring, and Fiber (specifically, the FDDI protocol) over different types of cabling. By repeating or amplifying signals where necessary, they enable the use of high quality UTP cabling in virtually every situation.

Hubs have evolved to provide tremendous flexibility for the design of the physical LAN topologies in large office buildings or plants. Various design strategies are now available. They are also an effective vehicle to put management intelligence throughout the LANs in a corporation, allowing control and monitoring capabilities from a network management center.

Newer token-passing protocols, such as Fiber Distributed Data Interface (FDDI) and Copper Distributed Data Interface (CDDI), will increase in use as higher performance LANs (particularly backbone LANs) are required. CDDI can be implemented on the same LAN cable as Ethernet and Token Ring if the original selection and installation are done carefully according to industry recommendations. FDDI usually appears first as the LAN-to-LAN bridge between floors in large buildings.

Wireless LANs offer an alternative to cabling. Instead of cabling, these LANs use the airwaves as the communications medium. Motorola provides a system-Altair-that supports standard Ethernet transmission protocols and cards. The Motorola implementation cables workstations together into microcells using standard Ethernet cabling. These microcells communicate over the airwaves to similarly configured servers. Communications on this frequency do not pass through outside walls, so there is little problem with interference from other users.

Wireless LANs are attractive when the cost of installing cabling is high. Costs tend to be high for cabling in old buildings, in temporary installations, or where workstations move frequently. NCR provides another implementation of wireless LAN technology using publicly accessible frequencies in the 902-MHz to 928-MHz band.

NCR provides proprietary cards to provide the communications protocol. This supports lower-speed communications that are subject to some interference, because so many other devices, such as remote control electronic controllers (like a VCR controller) and antitheft devices, use this same frequency.

It is now a well-accepted fact that LANs are the preferred vehicle to provide overall connectivity to all local and distant servers. WAN connectivity should be provided through the interconnection of the LANs. Router and bridges are devices that perform that task. Routers are the preferred technology for complex network topologies, generating efficient routing of data packets between two systems by locating and using the optimal path. They also limit the amount of traffic on the WAN by efficiently filtering and by providing support for multiple protocols across the single network.

WAN bandwidth for data communications is a critical issue. In terminal-to-host networks, traffic generated by applications could be modeled, and the network would then be sized accordingly, allowing for effective use of the bandwidth. With LAN interconnections, and applications that enable users to transfer large files (such as through e-mail attachments) and images, this modeling is much harder to perform. WAN services that have recently emerged, such as Frame Relay, SMDS (Switched Multimegabit Data Service), and imminent ATM (Asynchronous Transfer Mode) services, enable the appropriate flexibility inherently required for these applications.

Frame Relay uses efficient statistical multiplexing to provide shared network resources to users. Each access line is shared by traffic destined for multiple locations. The access line speed is typically sized much higher than the average throughput each user is paying for. This enables peak transmissions (such as when a user transmits a large file) that are much faster because they use all available bandwidth.

SMDS is a high-speed service that uses cell relay technology, which enables data, voice, and video to share the same network fabric. Available from selected RBOCs as a wide-area service, it supports high speeds well over 1.5 Mbps.

ATM is an emerging standard and set of communication technologies that span both the LAN and the WAN to create a seamless network. It provides the appropriate capabilities to support all types of voice, data, and video traffic. Its speed is defined to be 155 Mbps, with variations and technologies that may enable it to run on lower speed circuits when economically appropriate. It will operate both as a LAN and a WAN technology, providing full and transparent integration of both environments.

ATM will be the most significant connectivity technology after 1995. ATM provides the set of services and capabilities that will truly enable the "computing anywhere" concept, in which the physical location of systems and data is made irrelevant to the user. It also provides the network managers with the required flexibility to respond promptly to business change and new applications.

Interoperability between distributed systems is not guaranteed by just providing network-based connectivity. Systems need to agree on the end-to-end handshakes that take place while exchanging data, on session management to set up and break conversations, and on resource access strategies. These are provided by a combination of network protocols such as Novell's IPX/SPX, NetBIOS, TCP/IP, and remote process interoperability technologies, such as RPC technology from Sun, Netwise, Sybase, Oracle, IBM's APPC, CPIC, and Named Pipes.

Network Management is an integral part of every network. The Simple Network Management Protocol (SNMP) is a well-accepted standard used to manage LANs and WANs through the management capabilities of hubs, routers, and bridges. It can be extended to provide basic monitoring performance measurements of servers and workstations. Full systems management needs much more functionality than SNMP can offer. The OSI management protocol, the Common Management Information Protocol (CMIP), which has the flexibility and capability to fully support such management requirements, will likely compete with an improved version of SNMP, SNMP V2.

5.2 Open Systems Interconnect

The existence of heterogeneous LAN environments in large organizations makes interoperability a practical reality. Organizations need and expect to view their various workgroup LANs as an integrated corporate-wide network. Citicorp, for example, is working to integrate its 100 independent networks into a single global net.[1] The OSI model provides the framework definition for developers attempting to create interoperable products.[2] Because many products are not yet OSI-compliant, there often is no direct correspondence between the OSI model and reality.

The OSI model defines seven protocol layers and specifies that each layer be insulated from the other by a well-defined interface.

5.2.1 Physical Layer

The physical layer is the lowest level of the OSI model and defines the physical and electrical characteristics of the connections that make up the network. It includes such things as interface specifications as well as detailed specifications for the use of twisted-pair, fiber-optic, and coaxial cables. Standards of interest at this layer for client/server applications are IEEE 802.3 (Ethernet), and IEEE 802.5 (Token Ring) that define the requirements for the network interface card (NIC) and the software requirements for the media access control (MAC) layer. Other standards here include the serial interfaces EIA232 and X.21.

5.2.2 Data Liık Layer

The data link layer defines the basic packets of data expected to enter or leave the physical network. Bit patterns, encoding methods, and tokens are known to this layer. The data link layer detects errors and corrects them by requesting retransmission of corrupted packets or messages. This layer is actually divided into two sublayers: the media access control (MAC) and the logical link control (LLC). The MAC sublayer has network access responsibility for token passing; collision sensing, and network control. The LLC sublayer operates above the MAC and sends and receives data packets and messages.

Ethernet, Token Ring, and FDDI define the record format of the packets (frames) being communicated between the MAC layer and Network layer. The internal formats are different and without conversion workstations cannot interoperate with workstations that operate with another definition.

5.2.3 Network Layer

The network layer is responsible for switching and routing messages to their proper destinations. It coordinates the means for addressing and delivering messages. It provides for each system a unique network address, determines a route to transmit data to its destination, segments large blocks of data into smaller packets of data, and performs flow control.

5.2.4 Transport Layer

When a message contains more than one packet, the transport layer sequences the message packets and regulates inbound traffic flow. The transport layer is responsible for ensuring end-to-end error-free transmission of data. The transport layer maintains its own addresses that get mapped onto network addresses. Because the transport layer services process on systems, multiple transport addresses (origins or desıination) can share a single network address.

5.2.5 Session Layer

The session layer provides the services that enable applications running at two processors to coordinate their communication into a single session. A session is an exchange of messages-a dialog between two processors. This layer helps create the session, inform one workstation if the other drops out of the session, and terminate the session on request.

5.2.6 Presentation Layer

The presentation layer is responsible for translating data from the internal machine form of one processor in the session to that of the other.

5.2.7 Application Layer

The application layer is the layer to which the application on the processor directly talks. The programmer codes to an API defined at this layer. Messages enter the OSI protocol stack at this level, travel through the layers to the physical layer, across the network to the physical layer of the other processor, and up through the layers into the other processor application layer and program.

5.3 Communications Interface Technology

Connectivity and interoperability between the client workstation and the server are achieved through a combination of physical cables and devices, and software that implements communication protocols.

5~3.1 LAN Cabling

One of the most important and most overlooked parts of LAN implementation today is the physical cabling plant. A corporation's investment in cabling is significant. For most though, it is viewed strictly as a tactical operation, a necessary expense. Implementation costs are too high, and maintenance is a nonbudgeted, nonexistent process. The results of this shortsightedness will be seen in real dollars through the life of the technology. Studies have shown that over 65 percent of all LAN downtime occurs at the physical layer.

It is important to provide a platform to support robust LAN implementation, as well as a system flexible enough to incorporate rapid changes in technology. The trend is to standardize LAN cabling design by implementing distributed star topologies around wiring closets, with fiber between wiring closets. Desktop bandwidth requirements can be handled by copper (including CDDI) for several years to come; however, fiber between wiring closets will handle the additional bandwidth requirements of a backbone or switch-to-switch configuration.

Obviously, fiber to the desktop will provide extensive long-term capabilities; however, because of the electronics required to support various access methods in use today, the initial cost is significant. As recommended, the design will provide support for Ethernet, 4M and 16M Token Ping, FDDI, and future ATM LANs.

Cabling standards include RG-58 *AIU* coaxial cable (thin-wire 10Base2 Ethernet), IBM Type 1 (shielded, twisted pair for Token Ring), unshielded twisted pair (UTP for lOBaseT Ethernet or Token Ring) and Fiber Distributed Data Interface (FDDI for 1OBaseT or Token Ring). Motorola has developed a wireless Ethernet LAN product-Altair-that uses 18-GHz frequencies. NCR's WaveLAN provides low-speed wireless LAN support.

Wireless LAN technology is useful and cost-effective when the cost of cable installation is high. In old buildings or locations where equipment is frequently moved, the cost of running cables may be excessive. In these instances wireless technology can provide an attractive alternative. Motorola provides an implementation that uses standard Ethernet NICs connecting a group of closely located workstations together with a transmitter. The transmitter communicates with a receiver across the room to provide the workstation server connection. Recent reductions in the cost of this technology make it attractive for those applications where the cost of cabling is more than $250 per workstation.

Wireless communication is somewhat slower than wired communication. Industry tests indicate a performance level approximately one-half that of wired 10-Mbps UTP Ethernet. NCR's alternative wireless technology, WaveLAN, is a slow-speed implementation using proprietary communications protocols and hardware. It also is subject to interference by other transmitters, such as remote control electronics, antitheft equipment, and point-of-sale devices.

5.3.2 Ethernet IEEE 802.3

Ethernet is the most widely installed network topology today. Ethernet networks have a maximum throughput of 10 Mbps. The first network interface cards (NICs) developed for Ethernet were much cheaper than corresponding NICs developed by IBM for Token Ring. Until recently, organizations who used non-IBM minicomputer and workstations equipment had few options other than Ethernet. Even today in a heterogeneous environment, there are computers for which only Ethernet NICs are available.

The large market for Ethernet NICs and the complete definition of the specification have allowed over 100 companies to produce these cards.3 Competition has reduced the price to little more than $100 per unit.

lOBaseT Ethernet is a standard that enables the implementation of the Ethernet protocol over telephone wires in a physical star configuration (compatible with phone wire installations). Its robustness, ease of use, and low cost driven by hard competition have made 1OBaseT the most popular standards-based network topology. Its pervasiveness is unrivaled: In 1994, new laptop computers will start to ship with 1OBaseT built in. IBM is now fully committed to support Ethernet across its product line.

5.3.3 Token Ring IEEE 802.5

IBM uses the Token Ring LAN protocol as the standard for connectivity in its products. In an environment that is primarily IBM hardware and SNA connectivity, Token Ring is the preferred LAN topology option. IBM's Token Ring implementation is a modified ring configuration that provides a high degree of reliability since failure of a node does not affect any other node. Only failure of the hub can affect more than one node. The hub isn't electric and doesn't have moving parts to break; it is usually stored in a locked closet or other physically secure area.

Token Ring networks implement a wire transmission speed of 4 or 16 Mbps. Older NICs will support only the 4-Mbps speed, but the newer ones support both speeds. IBM and Hewlett-Packard have announced a technical alliance to establish a single 1OOMbps standard for both Token Ring and Ethernet networks. This technology, called lOOVG-AnyLAN, will result in low-cost, high-speed network adapter cards that can be used in PCs and servers running on either Token Ring or Ethernet LANs. The first AnyLAN products are expected in early 1994 and will cost between $250 and $3 50 per port. IBM will be submitting a proposal to make the 1OOVG-AnyLAN technology a part of IEEE's 802.12 (or 1OOBase-VG) standard, which currently includes only Ethernet. A draft IEEE standard for the technology is. expected by early 1994.

lOOVG-AnyLAN is designed to operate over a variety of cabling, including unshielded twisted pair (Categories 3, 4, or 5), shielded twisted pair, and FDDI.

The entire LAN operates at the speed of the slowest NIC. Most of the vendors today, including IBM and Synüptics, support 16 Mbps over unshielded twisted-pair cabling (UTP). This is particularly important for organizations that are committed to UTP wiring and are considering the use of the Token Ring topology.

5.3.4 Fiber Distributed Data Interface

The third prevalent access method for Local Area Networks is Fiber Distributed Data Interface (FDDI). FDDI provides support for 100 Mbps over optical fiber, and offers improved fault tolerance by implementing logical dual counter rotating rings. This is effectively running two LANs. The physical implementation ofFDDI is in a star configuration, and provides support for distances ofup to 2 km between stations.

FDDI is a next-generation access method. Although performance, capacity, and throughput are assumed features, other advantages support the use of FDDI in high-performance environments. FDDI's dual counter-rotating rings provide the inherent capability of end-node fault tolerance. By use of dual homing hubs (the capability to have workstations and hubs connected to other hubs for further fault tolerance), highly critical nodes such as servers or routers can be physically attached to the ring in two distinct locations. Station Management Technology (SMT) is the portion ofthe standard that provides ring configuration, fault isolation, and connection management. This is an important part of FDDI, because it delivers tools and facilities that are desperately needed in other access method technologies.

There are two primary applications for FDDI: first as a backbone technology for interconnecting multiple LANs, and second, as a high-speed medium to the desktop where bandwidth requirements justify it.

Despite the rapid decrease in the cost of Token Ring and lOBaseT Ethernet cards, FDDI costs have been decreasing at a faster rate. The costs of installation are dropping as preterminated cable reaches the market. Northern Telecom is anticipating, with its FibreWorld products, a substantial increase in installed end-user fiber driven by the bandwidth demands of multimedia and the availability requirements of business critical applications.

5.3.5 Copper Distributed Data Interface

The original standards in the physical layer specified optical fiber support only. Many vendors, however, have developed technology that enables FDDI to run over copper wiring. Currently, there is an effort in the ANSI X3T9.5 committee to produce a standard for FDDI over Shielded Twisted Pair (IBM compliant cable), as well as Data grade unshielded twisted pair. Several vendors, including DEC, IBM, and SynOptics are shipping an implementation that supports STP and UTP.

5.3.6 Ethernet versus Token Ring versus FDDI

The Ethernet technique works well when the cable is lightly loaded but, because of collisions that occur when an attempt is made to put data onto a busy cable, the technique provides poor performance when the LAN utilization exceeds 50 percent. To recover from the collisions, the sender retries, which puts additional load on the network. Ethernet users avoid this problem by creating subnets that divide the LAN users into smaller groups, thus keeping a low utilization level.

Despite the widespread implementation of Ethernet, Token Ring installations are growing at a fast rate for client/server applications. IBM's commitment to Ethernet may slow this success, because Token-Ring will always cost more than Ethernet.

The analysis predicts a steady increase in planned Token Ring installations from 1988 until the installed base is equivalent in 1996. However, this analysis does not account for the emergence of a powerful new technology which has entered the marketplace in 1993, Asynchronous Mode, or ATM. It is likely that by 1996 ATM will dominate all new installations and will gradually replace existing installations by 1999.

Token Ring performance is slightly poorer on lightly loaded LANs but shows linear degradation as the load increases, whereas Ethernet shows exponential degradation after loading reaches 30 percent capacity.

5.3.7 Asynchronous Transfer Mode (ATM)

ATM has been chosen by CCITT as the basis for its Broadband Integrated Services Digital Network (B-ISDN) services. In the USA, an ANSI-sponsored subcommittee also is investigating ATM.

The integrated support for all types of traffic is provided by the implementation of multiple classes of service categorized as follows:

Constant Bit Rate (CBR): connection-oriented with a timing relationship between the source and destination, for applications such as 64 kbits voice or fixed bit rate video

Variable Bit Rate (VBR): connection-oriented with a timing relationship between the source and destination, such as variable bit rate video and audio

Bursty traffic: having no end-to-end timing relationship, such as computer data and LAN-to-LAN

ATM's capability to make the "computing aywhere" concept a reality is made possible because ATM eventually will be implemented seamlessly both in the LAN and in the WAN. By providing a single network fabric for all applications, ATM also gives network managers with the required flexibility to respond promptly to business change and new applications.

5.3.8 Hubs

One of the most important technologies in delivering LAN technology to mainstream information system architecture is the intelligent hub. Recent enhancements in the capabilities of intelligent hubs have changed the way LANs are designed. Hubs owe their success to the efficiency and robustness of the 1OBaseT protocol, which enables the implementation of Ethernet in a star fashion over Unshielded Twisted Pair. Now commonly used, hubs provide integrated support for the different standard topologies (such as Ethernet, Token-Ring, and FDDI) over different types of cabling. By repeating or amplifying signals where necessary, they enable the use of high-quality UTP cabling in virtually every situation.

These intelligent hubs provide the necessary functionality to distribute a structured hardware and software system throughout networks, serve as network integration and control points, provide a single platform to support all LAN topologies, and deliver a foundation for managing all the components of the network.

There are three different types of hubs. *Workgroup hubs* support one LAN segment and are packaged in a small footprint for small branch offices. *Wiring closet hubs* support multiple LAN segments and topologies, include extensive management capabilities, and can house internetworking modules such as routers or bridges. *Network center hubs,* at the high end, support numerous LAN connections, have a high-speed backplane with flexible connectivity options between LAN segments, and include fault tolerance features.

Hubs have evolved to provide tremendous flexibility for the design of the physical LAN topologies in large office buildings or plants. Various design strategies are now available.

The distributed backbone strategy takes advantage of the capabilities of the wiring closet hubs to bridge each LAN segment onto a shared backbone network. This method is effective in large plants where distances are important and computing facilities can be distributed.

The collapsed backbone strategy provides a cost-effective alternative that enables the placement of all LAN servers in a single room and also enables the use of a single high-performance server with multiple LAN attachments. This is particularly attractive because it provides an environment for more effective LAN administration by a central group, with all servers easily reachable. It also enables the use of high-capacity, fault-tolerant internetworking devices to bridge all LAN segments to form an integrated network.

Hubs are also an effective vehicle to put management intelligence throughout the LANs in a corporation, allowing control and monitoring capabilities from a Network Management Center. This is particularly important as LANs in branch offices become supported by a central group.

5.3.9 Internetworking Devices Bridges and Routers

Internetworking devices enable the interconnection of multiple LANs in an integrated network. This approach to networking is inevitably supplanting the terminal-to-host networks as the LAN becomes the preferred connectivity platform to all personal, workgroup, or corporate computing facilities.

Bridges provide the means to connect two LANs together-in effect, to extend the size of the LAN by dividing the traffic and enabling growth beyond the physical limitations of any one topology. Bridges operate at the data link layer of the OSI model, which makes them topology-specific. Thus, bridging can occur between identical topologies only (Ethernet-to-Ethernet, Token Ring-to-Token Ring). Source-Route Transparent bridging, a technology that enables bridging between Ethernet and Token-Ring LANs, is seldom used.

Although bridges may cost less, some limitations must be noted. Forwarding of broadcast packets can be detrimental to network performance. Bridges operate promiscuously, forwarding packets as required. In a large internetwork, broadcasts from devices can accumulate, effectively taking away available bandwidth and adding to network utilization. "Broadcast storms" are rarely predictable, and can bring a network completely to a halt. Complex network topologies are difficult to manage. Ethernet bridges implement a simple decision logic that requires that only a single path to a destination be active. Thus, in complex meshed topologies, redundant paths are made inoperative, a situation that rapidly becomes ineffective as the network grows.

Routers operate at the network layer of the OSI model. They provide the means to intelligently route traffic addressed from one LAN to another. They support the transmission of data between multiple standard LAN topologies. Routing capabilities and strategies are inherent to each network protocol. IP can be routed through the OSPF routing algorithm, which is different than the routing strategy for Novell's IPX/SPX protocol. Intelligent routers can handle multiple protocols; most leading vendors carry products that can support mixes of Ethernet, Token Ring, FDD!, and from 8 to 10 different protocols.

5.3.10 Transmission Control Protocol/Internet Protocol

Many organizations were unable to wait for the completion of the OSI middle-layer protocols during the 1980s. Vendors and users adopted the Transmission Control Protocol/Intemet Protocol (TCP/IP), which was developed for the United States military Defense Advanced Research Projects Agency (DARPA) ARPANET network. ARPANET was one of the first layered communications networks and established the precedent for successful implementation of technology isolation between functional components. Today, the Internet is a worldwide interconnected network of universities, research, and commercial establishments; it supports thirty million US users and fifty million worldwide users. Additional networks are connected to the Internet every hour of the day. In fact growth is now estimated at 15 percent per month. The momentum behind the Internet is tremendous.

The TCP/IP protocol suite is now being used in many commercial applications. It is particularly evident in internetworking between different LAN environments. TCP/IP is specifically designed to handle communications through "networks of interconnected networks." In fact, it has now become the de facto protocol for LAN-based Client/Server connectivity and is supported on virtually every computing platform. More importantly, most interprocess communications and development tools embed support for TCP/IP where multiplatform interoperability is required. It is worth noting that IBM has followed this growth and not only provides support for TCP/IP on all its platforms, but now enables the transport of its own interoperability interfaces (such as CPIC, APPC) on TCP/IP.

5.3.10.1 TCPIIP's Architecture

The TCP/IP protocol suite is composed of the following components: a network protocol (IP) and its routing logic, three transport protocols (TCP, UDP, and ICMP), and a series of session, presentation and application services. The following sections highlight those of interest.

5.3.10.2 Internet Protocol

IP represents the network layer and is equivalent to OSI's IP or X.25. A unique network address is assigned to every system, whether the system is connected to a LAN or a WAN. The system comes with its associated routing protocols and lower level functions such as network-to-physical address resolution protocols (ARP). Commonly used routing protocols include RIP, OSPF, IGRP, and Cisco's proprietary protocol. OSPF has been adopted by the community to be the standards-based preferred protocol for large networks.

5.3.10.3 Transport Protocols

TCP provides Transport services over IP. It is connection-oriented, meaning it requires a session to be set up between two parties to provide its services. It ensures end-to-end data transmission, error recovery, ordering of data, and flow control. TCP provides the kind of communications that users and programs expect to have in locally connected sessions.

UDP provides connectionless transport services, and is used in very specific applications that do not require end-to-end reliability such as that provided by TCP.

5.3.10.4 Telnet

Telnet is an application service that uses TCP. It provides terminal emulation services and supports terminal-to-host connections over an internetwork. It is composed of two different portions: a client entity that provides services to access hosts and a server portion that provides services to be accessed by clients. Even workstation operating systems such as OS/2 and Windows can provide telnet server support, thus enabling a remote user to log onto the workstation using this method.

5.3.10.5 File Transfer Protocol (FTP)

FTP uses TCP services to provide file transfer services to applications. FTP includes a client and server portion. Server FTP listens for a session initiation request from client FTP. Files may be transferred in either direction, and ASCII and binary file transfer is supported. FTP provides a simple means to perform software distribution to hosts, servers, and workstations.

5.3.11 Simple Network Management Protocol (SNMP)

SNMP provides intelligence and services to effectively manage an internetwork. It bas been widely adopted by hub, bridge, and router manufacturers as the preferred technology to monitor and manage their devices.

SNMP uses UDP to support communications between agents-intelligent software that runs in the devices-and the manager, which runs in the management workstation. Two basic forms of communications can occur: SNMP polling (in which the manager periodically asks the agent to provide status and performance data) and trap generation (in which the agent proactively notifies the manager that a change of status or an anomaly is occurring).

5.3.12 Network File System (NFS)

The NFS protocol enables the use of IP by servers to share disk space and files the same way a Novell or LAN Manager network server does. It is useful in environments in which servers are running different operating systems. However, it does not offer support for the same administration facilities that a NetWare environment typically provides.

5.3.13 Simple Mail Transfer Protocol (SMTP)

SMTP uses TCP connections to transfer text-oriented electronic mail among users on the same host or among hosts over the network Developments are under way to adopt a standard to add multimedia capabilities (MIME) .to SMTP. Its use is widespread on the Internet, where it enables any user to reach millions of users in universities, vendor organizations, standards bodies, and so on. Most electronic mail systems today provide some form of SMTP gateway to let users benefit from this overall connectivity.

5.3.13.1 TCP/IP and Internerworks

Interestingly, the interconnected LAN environment exhibits many of the same characteristics found in the environment for which TCP/IP was designed. In particular

Routing: Intemetworks need support for routing; routing is very efficient in TCP/IP environments with efficient protocols such as OSPF.

Connections versus Connectionless: LAN activity includes both; the TCP/IP protocol suite efficiently supports both within an integrated framework.

Administrative Load Sensitivity: A LAN administrative support is usually limited; contrary to IBM's SNA, TCP/IP environments contain a tremendous amount of dynamic capabilities, in which devices and networks are dynamically discovered, and routing tables are automatically maintained and synchronized.

Networks of Networks: TCP/IP provides extreme flexibility as the administrative appreach to the management of federations of networks. Taking advantage of its dynamic nature, it enables very independent management of parts of a network (if appropriate).

5.3.14 Vendor Products

One of the leading vendors providing TCP/IP support for heterogeneous LANs is FTP Software of Wakefield, Massachusetts, which has developed the Clarkson Packet Drivers. These drivers enable multiple protocols to share the same network adapter. This is particularly useful, if not necessary, for workstations to take advantage of file and print services of a NetWare server, while accessing a client/server application located on a UNIX or Mainframe server.

IBiv[and Digital both provide support for TCP/IP in all aspects of their products' interoperability. Even IBM's LU6.2/ APPC specification can now run over a TCP/IP network, taking advantage of the ubiquitous nature of the protocol. TCP/IP is widely implemented, and its market presence will continue to grow.

5.4 Interprocess Communication

At the top of the OSI model, interprocess communications (!PCs) define the format for application-level interprocess communications. In the client/server model, there is always a need for interprocess communications. !PCs take advantage of services provided by protocol stacks such as TCP/IP, LU6.2, Decnet or Novell's IPX/SPX. In reality, a great deal of IPC is involved in most client/server applications, even where it is not visible to the programmer. For example, a programmer programming using ORACLE tools ends up generating code that uses IPC capabilities embedded in SQL *net, which provide the communications between the client application and the server.

The use of IPC is inherent in multitasking operating environments. The various active tasks operate independently and receive work requests and send responses through the appropriate IPC protocols. To effectively implement client/server applications, !PCs are used that operate equivalently between processes in a single machine or across machine boundaries on a LAN or a WAN.

!PCs should provide the following services:

Protocol for coordinating sending and receiving of data between processes

Queuing mechanism to enable data to be entered asynchronously and faster than it is processed

Support for many-to-one exchanges (a server dealing with many clients)

Network support, location independence, integrated security, and recovery

Remote procedure support to invoke a remote application service

Support for complex data structures

Standard programming language interface

All these features should be implemented with,w;fütle code and excellent performance.

5.4.1 Peer-to-Peer Protocols

A peer-to-peer protocol is a protocol that supports communications between equals. This type of communication is required to synchronize the nodes involved in a client/server network application and to pass work requests back and forth.

Peer-to-peer protocols are the opposite of the traditional dumb terminal-to-host protocols. The latter are hierarchical setups in which all communications are initiated by the host. NetBIOS, APPC, and Named Pipes protocols all provide support for peer-to-peer processing.

5.4.2 NetBIOS

The Network Basic I/O System (NetBIOS) is an interface between the transport and session OSI layers that was developed by IBM and Sytek in 1984 for PC connectivity. NetBIOS is used by DOS and OS/2 and is commonly supported along with TCP/IP. Many newer UNIX implementations include the NetBIOS interface under the name RFC to provide file server support for DOS clients.

NetBIOS is the de facto standard today for portable network applications because of its IBM origins and its support for Ethernet, Token Ring, ARCnet, StarLAN, and serial port LANs, and its IBM origins.

The NetBIOS commands provide the following services:

General: Reset, Status, Cancel, Alert, and Unlink. The general services provide miscellaneous but essential administrative networking services.

Name: Add, Add Group, Delete, and Find. The naming services provide the capability to install a LAN adapter card with multiple logical names. Thus, a remote adapter can be referred to by a logical name such as Hall Justice, R601 rather than its burned-in address ofX'1234567890123456'.

Session: Call, Listen, Send, Chain Send, Send No-Ack, Receive, Receive Any, Hang Up, and Status. Sessions provide a reliable logical connection service over which a pair of network applications can exchange information. Each packet of information that gets exchanged over a session is given a sequence number, through which it is tracked and individually acknowledged. The packets are received in the order sent and blocked into user messages. Duplicate packets are detected and discarded by the sessions services. Session management adds approximately five percent overhead to the line protocol.

Datagram: Send, Send-Broadcast, Receive, and Receive-Broadcast. Datagrams provide a simple but unreliable transmission service, with powerful broadcast capabilities. Datagrams can be sent to a named location, to a selected group (multicast) or to all locations on the network (broadcast). There is no acknowledgment or tracking of the datagram. Applications requiring a guarantee of delivery and successful processing must devise their own schemes to support such acknowledgment.

5.4.3 Application Program-to-Program Communication

The application program-to-program communication (APPC) protocol provides the necessary IPC support for peer-to-peer communications across an SNA network. APPC provides the program verbs in support of the LU6.2 protocol. This protocol is implemented on all IBM and many other vendor platforms. Unlike NetBIOS or Named Pipes, APPC provides the LAN and WAN support to connect with an SNA network, that may interconnect many networks.

Standards for peer-to-peer processing have evolved and have been accepted by the industry. IBM defined the LU6.2 protocol to support the *handshaking* necessary for cooperative processing between intelligent processors. Most vendors provide direct support for LU6.2 protocols in their WAN and the OSI committees and have agreed to define the protocol as part of the OSI standard for peer-to-peer applications. A recently quoted comment, "The U.S. banking system would probably collapse if a bug were found in IBM's LU6.2," points out the prevalence of this technology in highly reliable networked transaction environments.4

Programmers have no need or right to work with LU6.2 directly. Even with the services provided by APis, such as APPC, the interface is unreasonably complex, and the opportunities for misuse are substantial. Vendors such as PeerLogic offer excellent interface products to enable programs to invoke the functions from COBOL or C. High-level languages, such as Windows 4GL, access network transparency products such as Ingres Net implemented in the client and server (or SQL*Net in Oracle's case).

These network products basically map layers five and six of the OSI model, generate LU6.2 requests directly to access remote SQL tables, and invoke remote stored procedures. These products include all the necessary code to handle error conditions, build parameter lists, maintain multiple sessions, and in general remove the complexity from the sight of the business application developer.

The power ofLU6.2 does not come without complexity. IBM has addressed this with the definition of a Common Programmers Interface for Communications (CPI-C). Application program-to-program communication (APPC) is the API used by application programmers to invoke LU6.2 services. Nevertheless, a competent VTAM systems programmer must be involved in establishing the connection between the LAN node and the SNA network. The APPC verbs provide considerable application control and flexibility. Effective use of APPC is achieved by use of application interface services that isolate the specifics of APPC from the developer. These services should be built once and reused by all applications in an installation.

APPC supports conversational processes and so is inherently half-duplex in operation. The use of parallel sessions provides the necessary capability to use the LAN/WAN connection bandwidth effectively. In evaluating LU6.2 implementations from different platforms, support for parallel sessions is an important evaluation criterion unless the message rate is low.

LU6.2 is the protocol of choice for peer-to-peer communications from a LAN into a WAN when the integrity of the message is important. Two-phase commit protocols for database update at distributed locations will use LU6.2 facilities to guarantee commitment of all or none of the updates. Because of LU6.2 support within DECNET and the OSI standards, developers can provide message integrity in a multiplatform environment.

5.4.4 Named Pipes

Named Pipes is an IPC that supports peer-to-peer processing through the provision of two-way communication between unrelated processes on the same machine or across the LAN. No WAN support currently exists. Named Pipes are an OS/2 IPC. The server creates the pipe and waits for clients to access it. A useful compatibility feature of Named Pipes supports standard OS/2 file service commands for access. Multiple clients can use the same named pipe concurrently. Named Pipes are easy to use, compatible with the file system, and provide local and remote support. As such, they provide the IPC of choice for client/server software that do not require the synchronization or WAN features of APPC.

Named Pipes provide strong support for many-to-one IPCs. They take advantage of standard OS/2 and UNIX scheduling and synchronization services. With minimal overhead, they provide the following:

A method of exchanging data and control information between different computers

Transparency of the interface to the network

API calls that facilitate the use of remote procedure calls (RPCs)

The use of an RPC across a named pipe is particularly powerful because it enableJ the requester to format a request into the pipe with no knowledge of the location of the server. The server is implemented transparently to the requester on "some" machine platform, and the reply is returned in the pipe. This is a powerful facility that is very easy to use. Named Pipes support should become widespread because Novell and OSF have both committed the necessary threads support.

One of the first client/server online transaction processing (OLTP) products on the market, Ellipse, is independent of any communications method, although it requires networking platforms to have some notion of sessions. One of the major reasons Cooperative Solutions chose OS/2 and LAN Manager as the first Ellipse platform is OS/2 LAN Manager's Named Pipes protocol, which supports sessions using threads within processes.

Ellipse uses Named Pipes for both client/server and interprocess communications on the server, typically, between the Ellipse application server and the database server, to save machine instructions and potentially reduce network traffic. Ellipse enables client/server conversations to take place either between the Ellipse client process and the Ellipse server process or between the Ellipse client process and the DBMS server, bypassing the Ellipse server process. In most applications, clients will deal with the DBMS through the Ellipse server, which is designed to reduce the number of request-response round trips between clients and servers by synchronizing matching sets of data in the client's working storage and the server DBMS.

Ellipse uses its sessions to establish conversations between clients and servers. The product uses a named pipe to build each client connection to SQL Server. Ellipse uses a separate process for Named Pipes links between the Ellipse server and the SQL Server product.

Ellipse also uses sessions to perform other tasks. For example, it uses a named pipe to emulate cursors in an SQL server database management system (DBMS}. Cursors are a handy way for a developer to step through a series of SQL statements in an application. (Sybase doesn't have cursors.) Ellipse opens up Named Pipes to emulate this function, simultaneously passing multiple SQL statements to the DBMS. An SQL server recognizes only one named pipe per user, so Ellipse essentially manages the alternating of a main session with secondary sessions.

On the UNIX side, TCP/IP with the Sockets Libraries option appears to be the most popular implementation. TCP/IP supports multiple sessions but only as individual

PfOCesses. Although UNIX implements low-overhead processes, there is still more overhead than incurred by the use of threads. LAN Manager for UNIX is an option, but few organizations are committed to using it yet.

Windows 3.x client support is now provided with the same architecture as the OS/2 implementation. The Ellipse Windows client will emulate threads. The Windows client requires an additional layer of applications flow-control logic to be built into the Ellipse environment's Presentation Services. This additional layer will not be exposed to applications developers, in the same way that Named Pipes were not exposed to the developers in the first version of the product.

The UNIX environment lacks support for threads in most commercial implementations. Cooperative Solutions hasn't decided how to approach this problem. Certainly, the sooner vendors adopt the Open Software Foundation's OSF/1 version of UNIX, which does support threads, the easier it will be to port applications, such as Ellipse, to UNIX.

The missing piece in UNIX thread support is the synchronization of multiple requests to the pipe as a single unit of work across a WAN. There is no built-in support to back off the effect of previous requests when a subsequent request fails or never gets invoked. This is the scenario in which APPC should be used.

5.4.5 Anonymous Pipes

Anonymous pipes is an OS/2 facility that provides an IPC for parent and child communications in a spawned-task multitasking environment. Parent tasks spawn child tasks to perform asynchronous processing. It provides a memory-based, fixed-length circular buffer, shared with the use of read and write handles. These handles are the OS/2 main storage mechanism to control resource sharing. This is a high-performance means of communication when the destruction or termination of a parent task necessitates the termination of all children and in-progress work.

5.4.6 Semaphores

Interprocess synchronization is required whenever shared-resource processing is being used. It defines the mechanisms to ensure that concurrent processes or threads do not interfere with one another. Access to the shared resource must be serialized in an agreed upon manner. *Semaphores* are the services used to provide this synchronization.

Semaphores may use disk or D-RAM to store their status. The disk is the most reliable and slowest but is necessary when operations must be backed out after failure and before restart. D-RAM is faster but suffers from a loss of integrity when there is a

,system failure that causes D-RAM to be refreshed on recovery. Many large operations use a combination of the two-disk to record start and end and D-RAM to manage in-flight operations.

5.4.7 Shared Memory

Shared memory provides IPC when the memory is allocated in a named segment. Any process that knows the named segment can share it. Each process is responsible for implementing synchronization techniques to ensure integrity of updates. Tables are typically implemented in this way to provide rapid access to information that is infrequently updated.

5.4.8 Queues

Queues provide IPC by enabling multiple processes to add information to a queue and a single process to remove information. In this way, work requests can be generated and performed asynchronously. Queues can operate within a machine or between machines across a LAN or WAN. File servers use queues to collect data access requests from many clients.

5.4.9 Dynamic Data Exchange

Through a set of APis, Windows and OS/2 provide calls that support the *Dynamic Data Exchange* (DDE) protocol for message-based exchanges of data among applications. DDE can be used to construct hot links between applications in which data can be fed from window to window without interruption intervention. For example, a hot link can be created between a 3270 screen session and a word processing document. Data is linked from the 3270 window into the word processing document. Whenever the key of the data in the screen changes, the data linked into the document changes too. The key of the 3270 screen transaction Account Number can be linked into a LAN database. As new account numbers are added to the LAN database, new 3270 screen sessions are created, and the relevant information is linked into the word processing document. This document then can be printed to create the acknowledgment letter for the application.

DDE supports warm links created so the server application notifies the client that the data has changed and the client can issue an explicit request to receive it. This type of link is attractive when the volume of changes to the server data are so great that the client prefers not to be burdened with the repetitive processing. If the server link ceases to exist at some point, use a warm rather than hot link to ensure that the last data iteration is available.

You can create request links to enable direct copy-and-paste operations between a server and client without the need for an intermediate clipboard. No notification of change in data by the server application is provided.

You define execute links to cause the execution of one application to be controlled by another. This provides an easy-to-use batch-processing capability.

DDE provides powerful facilities to extend applications. These facilities; available to the desktop user, considerably expand the opportunity for application enhancement by the user owner. Organizations that wish to integrate desktop personal productivity fools into their client/server applications should insist that all desktop products they acquire be DDE-capable.

5.4.10 Remote Procedure Calls

Good programmers have developed modular code using structured techniques and subroutine logic for years. Today, these subroutines should be stored "somewhere" and made available to everyone with the right to use them. RPCs provide this capability; they standardize the way programmers must write calls to remote procedures so that the procedures can recognize and respond correctly.

If an application issues a functional request and this request is embedded in an RPC, the requested function can be located anywhere in the enterprise the caller is authorized to access. Client/server connections for an RPC are established at the session level in the OSI stack. Thus, the RPC facility provides for the invocation and execution of requests from processors running different operating systems and using different hardware platforms from the caller's. The standardized request form provides the capability for data and format translation in and out. These standards are evolving and being adopted by the industry.

Sun RPC, originally developed by Netwise, was the first major RPC implementation. It is the most widely implemented and available RPC today. Sun mcludes this RPC as part of their Open Network Computing (ONC) toolkit. ONC provides a suite of tools to support the development of client/server applications.

The Open Software Foundation (OSF) has selected the Hewlett-Packard (HP) Apollo RPC to be part of its distributed computing environment (DCE). This RPC-based on Apollo's Network Computing System (NCS)-is now supported .by Digital Equipment Corporation, Microsoft, IBM, Locus Computing Corp., and Transarc. OSI also has proposed a standard for RPC-like functions called Remote Operation Service (ROSE). The selection by OSF likely will make the HP standard the de facto industry standard after 1994. Organizations wishing to be compliant with the OSF direction should start to use this RPC today.

Organizations that want to build applications with the capability to use RPCs can create an architecture as part of their systems development environment (SDE) to support the standard RPC when it is available for their platform. All new development should include calls to the RPC by way of a standard API developed for the organization. With a minimal investment in such an API, the organization will be ready to take advantage of the power of their RPC as it becomes generally available, with very little modification of applications required.

When a very large number of processes are invoked through RPCs, performance will become an issue and other forms of client/server connectivity must be considered. The preferred method for high-performance IPC involves the use of peer-to-peer messaging. This is not the store-and-forward messaging synonymous with e-mail but a process-to-process communications with an expectation of rapid response (without the necessity of stopping processing to await the result).

The Mach UNIX implementation developed at Carnegie Mellon is the first significant example of a message-based operating system. Its performance and functionality have been very attractive for systems that require considerable interprocess communications. The NeXT operating system takes advantage of this message-based IPC to implement an object-oriented operating system.

The advantage of this process-to-process communication is evident when processors are involved in many simultaneous processes. It is evident how servers will use this capability; however, the use in the client workstation, although important, is less clear. New client applications that use object-level relationships between processes provide considerable opportunity and need for this type of communication. For example, in a text-manipulation application, parallel processes to support editing, hyphenation, pagination, indexing, and workgroup computing may all be active on the client workstation. These various tasks must operate asynchronously for the user to be effective.

A second essential requirement is object-level linking. Each process must view the information through a consistent model to avoid the need for constant conversion and subsequent incompatibilities in the result.

NeXTStep, the NeXT development environment and operating system, uses PostScript and the Standard Generalized Markup Language (SGML) to provide a consistent user and application view of textual information. IBM's peer-to-peer specification LU6.2 provides support for parallel sessioning thus reducing much of the overhead associated with many RPCs, that is, the establishment of a session for each request. IBM has licensed this technology for use in its implementation of OSFil.

RPC technology is here and working, and should be part of every client/server implementation. As we move into OLTP and extensive use of multitasking workgroup environments, the use of message-based !PCs will be essential. DEC's implementation is called DECmessageQ and is a part of its Application Control Architecture. The OSF Object Management Group (OMG) has released a specification for an object request broker that defines the messaging and RPC interface for heterogeneous operating systems and networks. The OMG specification is based on several products already in the marketplace, specifically HP's NewWave with Agents and the RPCs from HP and Sun. Organizations that want to design applications to take advantage of these facilities as they become available can gain considerable insight by analyzing the NewWave agent process. Microsoft has entered into an agreement with HP to license this software for inclusion in Windows NT.

5.4.11 Object Linking and Embedding

OLE is designed to let users focus on data-including words, numbers, and graphics-rather than on the software required to manipulate the data. A document becomes a collection of objects, rather than a file; each object remembers the software that maintains it. Applications that are OLE-capable provide an API that passes the description of the object to any other application that requests the object.

5.5 Wide Area Network Technologies

WAN bandwidth for data communications is a critical issue. In terminal-to-host networks, traffic generated by applications could be modeled, and the network would then be sized accordingly, enabling effective use of the bandwidth. With LAN interconnections and applications that enable users to transfer large files (such as through e-mail attachments) and images, this modeling is much harder to perform.

"Bandwidth-on-demand" is the paradigm behind these emerging technologies. Predictability of applications requirements is a thing of the past. As application developers get tooled for rapid application development and as system management facilities enable easy deployment of these new applications, the lifecycle of network redesign and implementation is dramatically shortened. In the short term, the changes are even more dramatic as the migration from a host-centric environment to a distributed client/server environment prevents the use of any past experience in "guessing" the actual network requirements.

Network managers must cope with these changes by seeking those technologies that will let them acquire bandwidth cost effectively while allowing flexibility to serve these new applications. WAN services have recently emerged that address this issue by providing the appropriate flexibility inherently required for these applications.

Distance-insensitive pricing seems to emerge as virtual services are introduced. When one takes into account the tremendous amount of excess capacity that the carriers have built into their infrastructure, this is not as surprising as it would seem. This will enable users and systems architects to become less sensitive to data and process placement when designing an overall distributed computing environment.

5.5.1 Frame Relay

Frame Relay network services are contracted by selecting two components: an access line and a *committed information rate* (CIR). This CIR speed is the actual guaranteed throughput you pay for. However, Frame Relay networks enable you, for example, to exceed this throughput at certain times to allow for efficient file transfers.

Frame Relay networks are often qualified as *virtual private networks.* They share a public infrastructure but implement virtual circuits between the senders and the receivers, similar to actual circuits. It is therefore a connection-oriented network. Security is provided by defining closed user groups, a feature that prevents devices from setting up virtual connections to devices they are not authorized to access.

5.5.2 Switched Multimegabit Data Service (SMDS)

SMDS is a high-speed service based on cell relay technology, using the same 53-byte cell transmission fabric as ATM. It also enables mixed data, voice, and video to share the same network fabric. Available from selected RBOCs as a wide-area service, it supports high speeds well over 1.5 Mbps, and up to 45 Mbps.

SMDS differs from Frame Relay in that it is a connectionless service. Destinations and throughput to those destination do not have to be predefined. Currently under trial by major corporations, S:MDS-at speeds that match current needs of customers-is a precursor to ATM services.

5.5.3 ATM in the Wide Area Network

The many advantages of ATM were discussed earlier in the chapter. Although not available as a service from the carriers, ATM will be soon be possible if built on private infrastructures.

Private networks have traditionally been used in the United States for high-traffic networks with interactive performance requirements. Canada and other parts of the world have more commonly used public X.25 networks, for both economic and technical reasons. With the installation of digital switching and fiber-optic communication lines, the telephone companies now find themselves in a position of dramatic excess capacity. In reality, it costs the telephone company to provide the service, initiate the call, and bill for it. There is no particular difference in the cost for distance and little in the cost for capacity. British Telecom has recently started offering a service with distance-insensitive pricing.

LANs provide a real opportunity to realize these savings. Every workstation on the LAN shares access to the wide-area facilities through the router or bridge. If the router has access to a Tl or T3 circuit, it can provide service on demand to any of the workstations on the LAN. This means that a single workstation can use the entire T1 for the period needed to transmit a document or file.

The time to transmit a character screen image is only 0.3 seconds with the 64-Kbps circuit. Therefore, increasing the performance of this transmission provides no benefit. If the transmission is a single-page image, such as a fax, the time to transmit is 164 seconds. This is clearly not an interactive response. Using a Tl circuit, the time reduces to only 5.9 seconds, and with a T3, to 0.2 seconds. If this image is in color, the times are 657 seconds compared to 23.5 and 0.8 seconds. In a client/server database application where the answer set to a query might be 10M, the time to transmit is 1,562 seconds (compared to 55.8 and 1.99 seconds).

When designing the architecture of the internetwork, it is important to take into account the communications requirements. This is not just an issue of total traffic, but also of instantaneous demand and user response requirements. ATM technologies will enable the use of the same lines for voice, data, or video communications without preallocating exclusive portions of the network to each application.

5.5.4 Integrated Services Digital Network

ISDN is a technology that enables digital communications to take place between two systems in a manner similar to using dial-up lines. Connections are established over the public phone network, but they provide throughput of up to 64 Kbps. ISDN has two basic components:

B-Channel: These two channels (hence the name of 2B+D for basic rate ISDN) provide communication services for either voice or data switched service. Data can be transmitted in any communications protocol.

D-Channel Signaling: This channel is used by the terminal equipment to control call setup and disconnection. It is much more efficient than call control of a dial-up line; the time required to set up a call is typically less than three seconds.

5.5.4.1 ISDN Applications

ISDN can provide high quality and performance services for remote access to a LAN. Working from the field or at home through ISDN, a workstation user can operate at 64 Kbps to the LAN rather than typical modem speeds of only 9.6 Kbps. Similarly, workstation-to-host connectivity can be provided through ISDN at these speeds. Help desk support often requires the remote help desk operator to take control of or share access with the user workstation display. GUI applications transmit megabits of data to and from the monitor. This is acceptable in the high-performance, directly connected implementation usually found with a LAN attached workstation; but this transmission is slow over a communications link.

Multimedia applications offer considerable promise for future use of ISDN. The capability to simultaneously send information over the same connection enables a telephone conversation, a video conference, and integrated workstation-to-workstation communications to proceed concurrently. Faxes, graphics, and structured data all can be communicated and made available for all participants in the conversation.

5.6 Network Management

When applications reside on a single central processor, the issues of network management assume great importance but often can be addressed by attentive operations staff With the movement to client/server applications, processors may reside away from this attentiveness.

If the data or application logic necessary to run the business resides at a location remote from the "glass house" central computer room, these resources must be visible to some network managers. The provision of a network control center (NCC) to manage all resources in a distributed network is the major challenge facing most large organizations today. The range of services is much greater than services traditionally implemented in terminal connected host applications. Many large organizations view this issue as the most significant obstacle to successful rollout of client/server applications.

The key layers in the management system architecture:

Presentation describes the management console environment and the tools used there.

Reduction refers to distributed intelligence, which acts as an intermediary for the network management interface. Reduction enables information to be consolidated and filtered, allowing the presentation service to delegate tasks through the use of an emerging distributed program services such as RPC, DME, or SMP. These provide the following benefits: response to problems and alerts can be executed locally to reduce latency and maintain availability, distributed intelligence can better serve a local environment-because smaller environments tend to be more homogeneous and such intelligence can be streamlined to reflect local requirements, scalability with regards to geography and political or departmental boundaries allows for local control and bandwidth optimization, reduction in management traffic overhead (because SNMP is a polling protocol), and placing distributed facilities locally reduced the amount of polling over a more expensive wide-area internet.

Gathering of information is done by device agents. Probably the greatest investment in establishing a base for the management network is through device management. Device management can represent the smallest piece of information, which may be insignificant in the overall picture. However, as network management tools evolve, the end result will be only as good as the information provided. These device agents provide detailed diagnostics, detailed statistics and precise control

OSF defines many of the most significant architectural components for client/server computing. The OSF selection of HP's Openview, combined with IBM's commitment to OSF's DME with its Netview/6000 product, ensures that we will see a dominant standard for the provision of network management services. There are five key OSI management areas:

Fault management

Performance management

Inventory management

Accounting management

Configuration management

The current state of distributed network and systems management illustrate serious weaknesses when compared to the management facilities available in the mainframe world today. With the adoption of Openview as the standard platform and including products such as Remedy Corporation's Action Request System for problem tracking/process automation, Tivoli's framework for system administration, management and security, and support applications from vendors such as Openvision, it is possible to implement effective distributed network and systems management today. The required integration will create more difficulties than mainframe operations might.

Standards organizations and the major vendors provide their own solution to this challenge. There is considerable truth in the axiom that "the person who controls the network controls the business." The selection of the correct management architecture for an organization is not straightforward and requires a careful analysis of the existing and planned infrastructure. Voice, data, application, video, and other nonstructured data needs must all be considered.

--- 6 ---

Client/Server Systems Development

--- Software ---

6.1 Executive Summary

If the selling price of automobiles had kept with the selling price of computer hardware, in 1992 dollars, a Geo would sell for $500. If the productivity improvement of· telephone operators had kept pace with the productivity improvement in systems development, 60 percent of the adult U.S. population would need to work as telephone operators to handle the current volume of calls compared to the volume of the 1920s.

An Index Group survey found that up to 90 percent of information technology (IT) departments' budgets are spent maintaining and enhancing existing systems. I This maintenance and enhancement continues to be done using old, inefficient, and undisciplined processes and technology. Figure 6.1 documents the change in maintenance effort measured in Fortune 1000 companies from the 1970s until today. As the number of installed systems increases, organizations find more of their efforts being invested in maintenance. Ed Yourdon claims that the worldwide software asset base is in excess of 150 billion lines of code. Most of this code was developed in the 1960s and 1970s with older technologies. Thus, this code is unstructured and undocumented, leading to what the Gartner Group is calling the "Maintenance Crisis." We simply must find more effective ways to maintain systems.

Business Process Reengineering (BPR) techniques help organizations achieve competitive advantage through substantive improvements in quality, customer service and costs. BPR must be aligned with technology strategy to be effective. Organizations must use technology to enable the business change defined by the BPR effort. In too many organizations technology is inhibiting change. Many CIOs are finding that their careers are much shortened when they discover that the business strategy identified by their organization cannot be realized because the technical architecture employed lacks the openness to support the change.

Senior executives look for new applications of technology to achieve business benefit. New applications must be built, installed, and made operational to achieve the benefits. Expenses incurred in maintenance and enhancement are not perceived to produce value. Yet, most measurements show that 66 percent of the cost of a system is incurred after its initial production release during the maintenance and enhancement phases. In this period of tight budgets it is increasingly difficult to explain and justify the massive ongoing investment in maintenance of systems that do not meet the current need.

Our challenge is to change the expenditures from ongoing maintenance to new development. Buying off-the-shelf application solutions frequently will meet the need. However, unless the packaged solution perfectly matches the needs of the organization, additional and expensive maintenance will be required to modify the package to make it fit.

Clearly, the solution is to design and build systems within a systems development environment (SDE). Applications and systems within an SDE are built to be maintained and enhanced. The flexibility to accept enhancements is inherent in the design. A methodology defines the process to complete a function. The use of a systems integration life cycle methodology ensures that the process considers the ramifications of all decisions made from business problem identification through and including maintenance and operation of the resultant systems. The changes implied by BPR and the movement from mainframe-centered development to client/server technology requires that you adopt a methodology that considers organizational transformation. Object-oriented technologies (OOTs) can now be used to define the necessary methodology and development environment to dramatically improve our ability to use technology effectively.

With effective use of 00 technologies productivity improvements of 10: 1 are being measured. Systems are being built with error rates that are one-third that of traditionally developed systems. The creation and reuse of objects supports the enterprise on the desk through the reuse of standard technology to support the user and developer. 00 technology allows business specialists to work as developers assembling applications by reusing objects previously constructed by more technical developers.

6.2 Factors Driving Demand for Application Software Development

Strategic planning, development, and follow-on support for applications software is a vital,-albeit expensive=process, that may yield enormous benefits in terms of cost savings, time to market for new products, customer satisfaction, and so on. There are opportunities to influence and compress application development planning time-through the use of an existing enterprise-wide architecture strategy or the adoption of a transformational outsourcing strategy. BPR and total quality management (TQM) programs demand software development and enhancements. A competitive market insists that companies demonstrate their value to a skeptical buyer through increasing the value of product and services.

6.3 Rising Technology Staff Costs

Coincident with the increasing demand for systems development, enrollment in university-level technology programs is declining, and the pool of available technical talent is shrinking relative to the exploding demand. As a result, technology personnel costs are rising much faster than inflation. In 1994, we see a 22-percent increase in demand for computer technologists. Many organizations find that technology professionals, in whom much organization specific application and technology knowledge has been invested, change jobs every three to five years. This multiplies the burden of reinvestment and retraining in organ-izations that are struggling to reduce costs. If organizations are to maximize their return on technology investments, they must develop a continuous learning program to ensure reuse of training programs, standard development procedures, developer tools and interfaces built for other systems.

6.3.1 Pressure to Build Competitive Advantage by Delivering Systems Faster

There is tremendous pressure on organizations to take advantage of new technology to build competitive advantage. This can be most easily accomplished by bringing innovative service offerings to market sooner than a competitor does. In most cases, new service offerings are required just to keep pace with competitors. The application backlog is horrific. Studies show that 80 to 90 percent of the traditional host-based MIS shop's staff time is devoted to maintaining or enhancing existing-often technically obsolete-applications. Some portion of the relatively small amount of time remaining is available for development of new applications.

For many organizations, implementing systems that not only increase efficiency and effectiveness but also transform fundamental processes to create a competitive advantage is absolutely essential to survival. For many companies, the prospects of global competition and uncertain recessionary times add fuel to the fire to succeed. Companies that cannot find inventive ways to refine their business process and streamline the value chain quickly will fall behind companies that can.

6.4 Need to Improve Technology Professionals' Productivity

The Index Group reports that the Computer-Aided Software Development (CASE) and other technologies that speed software development are cited by 70 percent of the tope IT executives surveyed as the most critical technologies to implement. The

CASE market is growing at a rate of30 percent per year, and Index's estimates predict it will be a $5 billion market by 1995, doubling from 1990 figures.

This new breed of software tools helps organizations respond more quickly by cutting the time it takes to create new applications and making them simpler to modify or maintain. Old methods, blindly automating existing manual procedures, can hasten a company's death knell. Companies need new, innovative mission-critical systems to be built quickly, with a highly productive, committed professional staff partnered with end-users during the requirements, design, and construction phases. The client/server development model provides the means to develop horizontal prototypes of an application as it is designed. The user will be encouraged to think carefully about the implications of design elements. The visual presentation through the workstation is much more real than the paper representation of traditional methods.

Yourdon reports that less than 20 percent of development shops in North America have a methodology of any kind, and even a lower percentage actually use the methodology. Input Research reports that internally developed systems are delivered on time and within budget about 1 percent of the time. They compare this result to those outsourced through systems integration professionals who use high-productivity environments, which are delivered on time and within budget about 66 percent of the time.

The use of a proven, formal methodology significantly increases the likelihood of building systems that satisfy the business need and are completed within their budgets and schedules. Yourdon estimates that 50 percent of errors in a final system and 75 percent of the cost of error removal can be traced back to errors in the analysis phase. CASE tools and development methodologies that define systems requirements iteratively with high and early user involvement have been proven to significantly reduce analysis phase errors.

6.5 Need for Platform Migration and Reengmeering of Existing Systems

Older and existing applications are being rigorously reevaluated and in some cases terminated when they don't pay off. A 15-percent drop in proprietary technology expenditures was measured in 1993 and this trend will continue as organizations move to open systems and workstation technology. BPR attempts to reduce business process cost and complexity by moving decision making responsibility to those individuals who first encounter the customer or problem. Organizations are using the client/server to bring information to the workplace of empowered employees.

The life of an application tends to be 5 to 15 years, whereas the life of a technology is much shorter-usually one to three years. Tremendous advances can be made by reengineering existing applications and preserving the rule base refined over the years while taking advantage of the orders-of-magnitude improvements that can be achieved using new technologies.

6.6 Need for a Common Interface Across Platforms

Graphical user interfaces (Gills) that permit a similar look and feel and front-end applications that integrate disparate applications are on the rise.

A 1991 *Information Week* survey of 157 IT executives revealed that ease of use through a common user interface across all platforms is twice as important as the next most important criteria as a purchasing criterion for software. This is the single-system image concept.

Of prime importance to the single-system image concept is that every user from every workstation have access to every application for which they have a need and right without regard to or awareness of the technology.

Developers should be equally removed from and unconcerned with these components. Development tools and APis isolate the platform specifics from the developer. When the single-systems image is provided, it is possible to treat the underlying technology platforms as a commodity to be acquired on the basis of price-performance without concern for specific compatibility with the existing application. Hardware, operating systems, database engines, communication protocols-all these must be invisible to the application developer.

6.7 Increase in Applications Development by Users

As workstation power grows and dollars-per-MIPS fall, more power is moving into the hands of the end user. The Index Group reports that end users are now doing more than one-third of application development; IT departments are functioning more like a utility. This is the result of IT department staff feeling the squeeze of maintenance projects that prevent programmers from meeting critical backlog demand for new development.

This trend toward application development by end-users will create disasters without a consistent, disciplined approach that makes the developer insensitive to the underlying components of the technology. End-user application developers also must understand the intricacies of languages and interfaces.

Object-oriented technologies embedded in SDE have regularly demonstrated to produce new development productivity gains of 2 to 1 and maintenance productivity improvements of 5 to 1 over traditional methods-for example, process-driven or data-driven design and development. More recently mature OO SDEs with a strong focus on object reusability are achieving productivity gains of 10 to 1 over traditional techniques.

Production-capable technologies are now available to support the development of client/server applications. The temptation and normal practice is to have technical staff read the trade press and select the best products from each category, assuming that they will combine to provide the necessary development environment. In fact, this almost never works. When products are not selected with a view as to how they will work together, they do not work together.

Thus, the best Online Transaction Processing (OLTP) package may not support YOUR best database. Your security requirements may not be met by any of your tools. Your applications perform well, but it may take forever to change them. Organizations must architect an environment that takes into account their particular priorities and the suite of products being selected. The selection of tools will provide the opportunity to be successful.

An enterprise-wide architecture strategy must be created to define the business vision and determine a transformation strategy to move from the current situation to the vision. This requires a clear understanding of industry standards, trends, and vendor priorities. Combining the particular business requirements with industry direction it is possible to develop a clear strategy to use technology to enable the business change. Without this architecture strategy, decisions will be made in a vacuum with little business input and usually little clear insight into technology direction.

The next and necessary step is to determine how the tools will be used within your organization. This step involves the creation of your SDE. Without the integration of an SDE methodology, organizations will be unable to achieve the benefits of client/server computing. Discipline and standards are essential to create platform-independent systems. With the uncertainty over which technologies will survive as standards, the isolation of applications from their computing platforms is an essential insurance policy.

6.8 Client/Server Systems Development Methodology

The purpose of a methodology is to describe a disciplined process through which technology can be applied to achieve the business objectives. Methodology should describe the processes involved through the entire life cycle, from BPR and systems planning through and including maintenance of systems in production. Most major systems integrators and many large in-house MIS groups have their own life cycle management methodology. Andersen Consulting, for example, has its Foundation, BSG has its Blueprint, and SHL Systemhouse has its own SHL Transform-the list goes on and on. These companies offer methodologies tuned for the client/server computing environment. However, every methodology has its own strengths, which are important to understand as part of the systems integration vendor selection process.

It is necessary to understand and adhere to the flow of information through the life cycle. This flow allows the creation and maintenance of the systems encyclopedia or electronic repository of data definitions, relationships, revision information, and so on. This is the location of the data models of all systems. The methodology includes a strict project management discipline that describes the deliverables expected from each stage of the life cycle. These deliverables ensure that the models are built and maintained. In conjunction with CASE tools, each application is built from the specifications in the model and in turn maintains the model's where-used and how-used relationships.

Table 6.1 details the major activities of each stage of the systems integration life cycle methodology. No activity is complete without the production of a formal deliverable that documents, for user signoff, the understanding gained at that stage. The last deliverable from each stage is the plan for the next stage.

Table 6.1. SILC phases and major activities.

SILCPhase	*Major Activities*
Systems Planning	Initiate systems planning
	Gather data
	Identify current situation
	Describe existing systems

	Define requirements
	Analyze applications and data architectures
	Analyze technology platforms
	Prepare implementation plan
Project Initiation	Screen request
	Identify relationship to long-range systems plan
	Initiate project
	Prepare plan for next phase
Architecture Definition	Gather data
	Expand the requirements to the next level o detail
	Conceptualize alternative solutions
	Develop proposed conceptual architecture
	Select specific products and vendors
Analysis	Gather data
	Develop a logical model of the new application system
	Define general information system requirements
	Prepare external system design
Design	Perform preliminary design
	Perform detailed design
	Design system test
	Design user aids
	Design conversion system
Development	Set up the development environment
	Code modules
	Develop user aids
	Conduct system test

Facilities Engineering	Gather data
	Conduct site survey
	Document facility requirements
	Design data center
	Plan site preparation
	Prepare site
	Plan hardware installation
	Install and test hardware
Implementation	Develop contingency procedures
	Develop maintenance and release procedures
	Train system users
	Ensure that production environment is ready
	Convert existing data
	Install application system
	Support acceptance test
	Provide warranty support
Post-implementation Support	Initiate support and maintenance services
	Support hardware and communication configuration
	Support software
	Perform other project completion tasks as appropriate

6.9 Project Management

Many factors contribute to a project's success. One of the most essential is establishing an effective project control and reporting system. Sound project control practices not only increase the likelihood of achieving planned project goals but also promote a working environment where the morale is high and the concentration is intense. This is particularly critical today when technology is so fluid and the need for isolating the developer from the specific technology is so significant.

The objectives of effective project management are to

Plan the project:

Define project scope

Define deliverables

Enforce methodology

Identify tasks and estimates

Establish project organization and staffing

Document assumptions

Identify client responsibilities

Define acceptance criteria

Define requirements for internal quality assurance review

Determine project schedulesand milestones

Document costs and payment terms

Manage and control project execution:

Maintain personal commitment

Establish regular status reporting

Monitor project against approved milestones

Follow established decision and change request procedures log and follow up on problems

Complete the project:

Establish clear, unambiguous acceptance criteria

Deliver a high-quality product consistent with approved criteria

Obtain clear acceptance of the product :

New technologies such as client/server place a heavy burden on the architecture definition phase. The lack of experience in building client/server solutions, combined with the new paradigm experienced by the user community, leads to considerable prototyping of applications. These factors will cause rethinking of the architecture. Such a step is reasonable and appropriate with today's technology. The tools for prototyping in the client/server platform are powerful enough that prototyping is frequently faster in determining user requirements than traditional modeling techniques were.

When an acceptable prototype is built, this information is reverse engineet¢q into the CASE tool's repository. Bachman's Information Systems' CASE products provide among the more powerful available tools to facilitate this process.

6.10 Architecture Definition

The purpose of the architecture definition phase in the methodology is to define the application architecture and select the technology platform for the application. To select the application architecture wisely, you must base the choice on an evaluation of the business priorities. Your organization must consider and weight the following criteria:

Cost of operation-How much can the organization afford to pay?

Ease of use-Are all system users well-trained, computer literate, and regular users? Are some occasional users, intimidated by computers, users with little patience, or familiar with another easy to use system? Will the system be used by the public in situations that don't allow for training or in which mistakes are potentially dangerous?

Response time-What is the real speed requirement? Is it less than 3 seconds 100 percent of the time? What is the impact if 5 percent of the time the response lag is up to 7 seconds?

Availability-What is the real requirement? Is it 24 hours per day, 7 days per week, or something less? What is the impact of outages? How long can they last before the impact changes?

Security-What is the real security requirement? What is the cost or impact of unauthorized access? Is the facility secure? Where else can this information be obtained?

Flexibility to change-How frequently might this application change? Is the system driven by marketing priorities, legislative changes, or technology changes?

Use of existing technology-What is the existing investment? What are the growth capabilities? What are the maintenance and support issues?

System interface-What systems must this application deal with? Are these internal or external? Can the systems being interfaced be modified?

These application architecture issues must be carefully evaluated and weighed from a business perspective. Only after completing this process can managers legitimately review the technical architecture options. They must be able to justify the technology selection in the way it supports the business priorities. There is always a desire to manage risk and a corresponding desire to use the best technology. A balance must be found between the two extremes of selecting something that fits the budget and is known to work versus the newest, best, and unproven option. Cost is always a consideration.

Once managers understand the application architecture issues, it becomes appropriate to evaluate the technical architecture options. Notice that staff are not yet selecting product, only architectural features. It is important to avoid selecting the product before purchasers understand the baseline requirements.

The following is a representative set of technical architecture choices:

Hardware (including peripherals)-Are there predefined standards for the organization? Are there environmental issues, such as temperature, dirt, and service availability? Distributed versus centralized-Does the organization have a requirement for one type of processing over the other? Are there organizational standards?

Network configuration-Does the organization have an existing network? Is there a network available to all the sites? What is the capacity of the existing network? What is the requirement of the new one?

Communications protocols-What does the organization use today? Are there standards that must be followed? System software-What is used today? Are there standards in place? What options are available in the locale and on the anticipated hardware and communications platforms? Database software-Is there a standard in the organization? What exists today?

Application development tools (for example, CASE)-What tools are in use today? What tools are available for the candidate platforms, database engine, operating system, and communications platforms?

Development environment-Does such an environment exist today? What standards are in place for users and developers? What other platform tools are being considered? What are the architectural priorities related to development?

Application software (make or buy, package selection, and so on)-Does the organization have a standard? How consistent is this requirement with industry-standard products? If there is a product, what platforms does it run on? Are these consistent with the potential architecture here? How viable is the vendor? What support is available? Is source code available? What are the application architecture requirements related to product acquisition?Human interface-What are the requirements? What is in place today? What are users expecting?

It illustrates the layering of technical architecture and applications architecture. One should not drive the other. It is unrealistic to assume that the application architects can ignore the technical platform, but they should understand the business priorities and work to see that these are achieved. Interfaces must isolate the technical platform from the application developers. These interfaces offer the assurance that changes can be made in the platform without affecting functioning at the application layer.

With the technical architecture well defined and the application architecture available for reference, you're prepared to evaluate the product options. The selection of the technology platform is an important step in building the SDE. There will be ongoing temptation and pressure to select only the "best products." However, the classification of "best product in the market, " as evaluated in the narrow perspective of its features versus those of other products in a category, is irrelevant for a particular organization. Only by evaluating products in light of the application and technical architecture in concert with all the products to be used together can you select the best product for your organization.

6.11 Systems Development Environment

Once your organization has defined its application and technical architectures and selected its tools, the next step is to define how you'll use these tools. Developers do not become effective system builders because they have a good set of tools; they become effective because their development environment defines how to use the tools well.

An SDE comprises hardware, software, interfaces, standards, procedures, and training that are selected and used by an enterprise to optimize its information systems support to strategic planning, management, and operations.

An architecture definition should be conducted to select a consistent technology platform.

Interfaces that isolate the user and developer from the specifics of the technical platform should be used to support the creation of a single-system image.

Standards and standard procedures should be defined and built to provide the applications with a consistent look and feel.Reusable components must be built to gain productivity and support a single-system image. Training programs must ensure that users and developers understand how to work in the environment.

IBM defined its SDE in terms of an application development cycle, represented by a product line it called AD/Cycle. Another way of looking at the. SOE.The SDE must encompass all phases of the systems development life cycle.and mustbeintegrated with the desktop. The desktop provides powerful additional tools for workstation users to become self-sufficient in many aspects of their information-gathering needs.

The most significant advantages are obtained from an SDE when a conscious effort is made to build reusable components. These are functions that will be used in many applications and will therefore improve productivity. Appendix A's case studies illustrate the benefits of projects built within the structure of an SDE. With the uncertainty surrounding product selection for client/server applications today, the benefits of using an SDE to isolate the developers from the technology are even more significant. These technologies will evolve, and we can build applications that are isolated from many of the changes. The following components should be included i~ any SDE established by an organization:

Built-in navigation-Every process uses the same methods to move among processes. For every process a default next process is identified, and all available processes are identified. This navigation definition is done by a business analyst and not the developer. Every user and every developer then views navigation in the same way.

Standardized screen design-Well-defined standards are in place for all function types, and these screens are generated by default based on the business process being defined. Users and developers become familiar with the types of screens used for help, add, change, delete, view, and table management functions.

Integrated help-A standardized, context-sensitive help facility should respond to the correct problem within the business process. No programmer development is required. The help text is provided by the end-user and analyst who understand how the system user will view the application. Help text is user maintainable after the system is in production.

Integrated table maintenance=-Tables are a program design concept that calls for common reference data, such as program error codes, printer control codes, and so on, to be stored in a single set of files or databases. A single table maintenance function is provided for all applications in the organization. Programmers and users merely invoke its services. Thus, all applications share standard tables.

Comprehensive security-A single security profile is maintained for each authorized user. Navigation is tied to security; thus, users only see options they are eligible to use. Every programmer and user see the same security facilities. Security profiles are maintained by an authorized user and usethe/tapl~mrrı~ıntenance facilities.

Automatic view maintenance-Screens are generated,}!~avigation is prescribed, and skeleton programs are generated based on the security profile and business requirements defined for a process. The developer does not have to write special code to extract data from the database. All access is generated based on the defined business processes and security.

Standard skeleton programs-An analyst answers a set of questions to generate a skeleton program for each business process. This feature includes standard functions that the programmer will require.

Every platform includes a set of services that are provided by the tools. This is particularly true in the client/server model, because many of the tools are new and take advantage of object-oriented development concepts. It is essential for an effective SDE to use the facilities and not to redevelop these because of elegance or ego.

The ACTS example shown in Appendix A uses this SDE architecture with Easel and Telon. Users and developers can move between these environments with minimal difficulty because there is such a high degree of commonalty in the look and feel and in the services provided. Development within the justice application (of which ACTS is a part) included the Software AG products, Easel, and Telon. The same developers were productive throughout because of the common architecture. This occurred despite the fact that portions of the application were traditional mainframe, portions were mixed workstation-to-mainframe programs, and portions were pure client/server.

The advantages of building an SDE and including these types of components are most evident in the following areas:

Rapid prototyping-The development environment generates skeleton applications with embedded logic for navigation, database views, security, menus, help, table maintenance, and standard screen builds. This framework enables the analyst or developer to sit with a user and work up a prototype of the application rapidly. In order to get business users to participate actively in the specification process, it is necessary to show them something real. A prototype is more effective for validating the process model than are traditional business modeling techniques. Only through the use of an SDE is such prototyping possible. Workstation technology facilitates this prototyping. The powerful GUI technology and the low cost of direct development at the workstation make this the most productive choice for developing client/server applications.

Rapid coding-Incorporating standard, reusable components into every program reduces the number of lines of custom code that must be written. In addition, there is a substantial reduction in design time, because much of the design employs reusable, standard services from the SDE. The prototype becomes the design tool.

Consistent application design-As mentioned earlier, much of the design is inherent in the SDE. Thus, by virtue of the prototype, systems have a common look and feel from the user's and the developer's perspectives. This is an essential component of the single-system image.

Simplified maintenance-The standard code included with every application ensures that when maintenance is being done the program will look familiar. Because more than 50 percent of most programs will be generated from reusable code, the maintenance programmer will know the modules and will be able to ignore them unless global changes are to be made in these service functions. The complexity of maintenance corresponds to the size of the code and the amount of familiarity the programmer has with the program source. The use of reusable code provides the programmer with a common look and much less new code to learn.

Enhanced performance-Because the reusable components are written once and incorporated in many applications, it is easier to justify getting the best developers to build the pieces. The ability to make global changes in reusable components means that when performance problems do arise, they can often be fixed globally with a single change.

6.12 Productivity Measures

It is difficult to accurately quantify productivity gains obtained by using one method versus another, because developers are not willing to build systems twice with two different teams with the same skill set. However, a limited number of studies have been done estimating the expected cost of developing and maintaining systems without a formal SDE compared to the actual results measured with an SDE. One such analysis studied U.S. competitiveness. The researchers determined that, on average, a Japanese development team produces 170 percent of the debugged lines of code per year that a U.S. development team does. Japanese literature describes the Japanese approach to building systems as very consistent with the SDE approach described here. The necessity for Japanese developers to deal with U.S. software and a Japanese script language user interface has taught them the value of software layers. This led naturally to the development of reusable software components. Measurements by the researchers of errors in systems developed by Japanese and United States development teams showed that the Japanese had only 44 percent of the errors measured in the U.S. code.

Japanese developers work in a disciplined style that emphasizes developing to standards and reuse of common components. Our experience with SDE-based development is showing a 100-percent productivity improvement for lines of debugged source code per work year for new development and a 400-percent productivity increase for maintenance of existing systems. It's easy to understand the new code improvement rate from the facts noted earlier, but it is not as clear why the maintenance improvement is so great.

A significant reason for better productivity appears to be the reduction in testing effort that results from fewer errors. It is difficult to make changes to a production application. The cost and effort involved in changing production code is dramatically greater than changes to a test system. Developers and testers are careful about changes to production products. If you eliminate half the errors, you not only have happier users but also a substantial reduction in effort to correct the problems. The ability to make global changes and the reduction in complexity that comes from the familiar environment also improve maintenance productivity.

6.13 CASE

CASE tools are built on an "enterprise model" of the processes to be automated; that is, systems integration and software development. This underlying enterprise model or "metarnodel" used by CASE is crucial to the tool's usefulness. Tools based on a poor model suffer from poor integration, are unable to handle specific types of information, require duplicate data entry, cannot support multiple analyst-developer teams, and are not flexible enough to handle evolving new techniques for specifying and building systems solutions. Tools with inadequate models limit their users' development capabilities.

All the leading CASE products operate and are used in a client/server environment. Intel 486-based workstations operating at 50:MHz or faster, with 16-24 Mbytes of memory and 250Mbyte hard disks and UNIX workstations of similar size are typically required. Thus, combining hardware and CASE software costs, CASE costs up to $20,000 per user workstation/terminal.

Unfortunately, a thorough review of the available CASE products shows that none adequately provide explicit support for development of client/server applications and GUis. This lack of support occurs despite the fact that they may operate as network-based applications that support development of host-based applications. There is considerable momentum to develop products that support the client/server model. The Bachman tools are in the forefront in this area because of their focus on support for business process reengineering. With many client/server applications being ported from a minicomputer or mainframe, the abilities to reuse the existing models and to reverse engineer the databases are extremely powerful and time-saving features.

It seems likely that no single vendor will develop the best integrated tool for the entire system's life cycle. Instead, in the probable scenario, developers mix the best products from several vendors. This scenario is envisioned by IBM in their AD/Cycle product line, by Computer Associates in their CA90 products, and by NCR in their Open Cooperative Computing series of products.

As an example, an organization may select Bachman, which provides the best reengineering and reusability components and the only true enterprise model for building systems solutions for their needs. This model works effectively in the LAN environment and supports object-oriented reuse of specifications. The organization then integrates the Bachman tools with ParcPlace Parts product for Smalltalk code generation for Windows, UNIX or OS/2 desktops and server applications and with Oracle for code generation in the UNIX, OS/2, and Windows NT target environment. The visual development environments of these products provide the screen painting, business logic relationship, and prototyping facilities necessary for effective systems development.

A more revolutionary development is occurring as CASE tools like the Bachman products are being integrated with development tools from other vendors. These development tools, used with an SDE, allow applications to be prototyped and then reengineered back into the CASE tool to create process and data models. With the power of GUI-based development environments to create and demonstrate application look and feel, the prototyping approach to rapid application design (RAD) is the only cost-effective way to build client/server applications today.

Users familiar with the ease of application development on the workstation will not accept paper or visual models of their application. They can only fully visualize the solution model when they can touch and feel it. This is the advantage of prototyping, which provides a "real touch and feel." Except in the earliest stages of solution conceptualization, the tools for prototyping must be created using the same products that are to be used for production development.

Not all products that fall into the CASE category are equally effective. For example, some experts claim that the information engineering products-such as Texas Instruments' product, IBP-attempt to be all things to all people. The criticism is that such products are constrained by their need to generate code efficiently from their models. As a result, they are inflexible in their approach to systems development, have primitive underlying enterprise models, may require a mainframe repository, perform poorly in a team environment, and provide a physical approach to analysis that is constrained by the supported target technologies (CICS/DB2 and, to a lesser extent, Oracle). Critics argue that prototyping with this class of tool requires developers to model an unreasonable amount of detail before they can present the prototype.

6.14 Object-Oriented Programming (OOP)

OOP is a disciplined programming style that incorporates three key characteristics: encapsulation, inheritance, and dynamic binding. These characteristics differentiate OOP from traditional structured programming models, in which data has a type and a structure, is distinct from the program code, and is processed sequentially. OOP builds on the concepts of reuse through the development and maintenance of class libraries of objects available for use in building and maintaining applications.

Encapsulation joins procedures and data to create an object, so that only the procedures are visible to the user; data is hidden from view. The purpose of encapsulation is to mask the complexity of the data and the internal workings of the object. Only the procedures (methods) are visible to the outside world for use.

Inheritance passes attributes to dependent objects, called descendants, or receives attributes from objects, called ancestors, on which the objects depend. For example, the family airplane includes all structures, whereas the descendant jet inherits all the properties of airplane and adds its own, such as being nonpropeller-driven; the child Fl4 inherits all the properties of airplane and jet and adds its own properties-speed greater than 1,400 mph and climbing rate greater than 50 feet per second.

Dynamic binding is the process whereby linking occurs at program execution time. All objects are defined at runtime, and their functions depend on the application's environment (state) at the time of program execution. For example, in a stock management application, the function called program trading can sell or buy, depending on a large range of economic variables that define the current state. These variables are transparent to the user who invokes the trade process.

Class library is a mature, tested library of reusable code that provides application-enabling code such as help management, error recovery, function key support, navigation logic, and cursor management. The class library concept is inherent to the SDE concept and-in combination with the standards and training fundamental-is inherent to the productivity and error reductions encountered in projects that use an SDE.

Object-oriented programming is most effective when the reusable components can be cut and pasted to create a skeleton application. Into this skeleton the custom business logic for this function is embedded. It is essential that the standard components use dynamic binding so that changes can be made and applied to all applications in the environment. This provides one of the major maintenance productivity advantages.

Certain programming languages are defined to be object-oriented. C++, Objective C, SmallTalk, MacApp, and Actor are examples. With proper discipline within an SDE it is possible to gain many of the advantages of these languages within the more familiar environments of COBOL and C. Because the state of development experience in the client/server world is immature, it's imperative for organizations to adopt the discipline of OOP to facilitate the reuse of common functions and to take advantage of the flexibility of global changes to common functions.

Objects are easily reused, in part because the interface to them is so plainly defined and in part because of the concept of inheritance. A new object can inherit characteristics of an existing object "type." You don't have to reinvent the wheel; you can just inherit the concept. Inheritance gives a concise and precise description of the world and helps code reusability, because every program is at the level in the "type hierarchy" at which the largest number of objects can share it. The resulting code is easier to maintain, extend, and reuse.

A significant new component of object-oriented development has been added with the capability to use server objects with RPC requests. During 1994, the introduction of CORBA compliant object stores will dramatically open the client/server paradigm to the "anything anywhere" dimension. Objects will be built and stored on an arbitrary server for use by any client or server anywhere. The earliest implementations of this model are provided by NeXT with its Portable Distributed Objects (PDO) and Suns Distributed Objects Everywhere (DOE) architecture.

And what about object-oriented database management system (OODBMS)? It combines the major object-oriented programming concepts of data abstraction, encapsulation, and type hierarchies with the database concepts of storage management, sharing, reliability, consistency, and associative retrieval.

When is an OODBMS needed, and when will an extended relational data-base management system (DBMS) do? Conventional database management products perform very well for many kinds of applications. They excel at processing large amounts of homogeneous data, such as monthly credit card billings. They are good for high-transaction-rate applications, such as ATM networks. Relational database systems provide good support for ad hoc queries in which the user declares what to retrieve from the database as opposed to how to retrieve it.

As we traverse the 1990s, however, database management systems are being called on to provide a higher level of database management. No longer will databases manage data; they must manage information and be the knowledge centers of the enterprise. To accomplish this, the database must be extended to

Provide a higher level of information integration

Store and retrieve all types of data: drawings, documents, fax, images, pictures, medical information, voice, and video

Many RDBMS products already handle binary large objects (BLOBs) in a single field of a relation. Many applications use this capability to store and provide SQL-based retrieval of digital laboratory data, images, text, and compound documents. Digital's Application Driven Database Systems (ADDS) have been established to enable its SQL to handle these complex and abstract data types more explicitly and efficiently.

But applications that require database system support are quickly extending beyond such traditional data processing into computer-aided design (CAD) and CASE, sophisticated office automation, and artificial intelligence. These applications have complex data structuring needs, significantly different data accessing patterns, and special performance requirements. Conventional programming methodologies are not necessarily appropriate for these applications and conventional data management systems may not be appropriate for managing their data.

Consider for a moment the factors involved in processing data in applications such as CAD, CASE, or generally in advanced office automation. The design data in a mechanical or electrical CAD database is heterogeneous. It consists of complex relationships among many types of data. The transactions in a CASE system don't lend themselves to transaction-per-second measurement; transactions can take hours or even days. Office automation applications deal with a hierarchical structure of paragraphs, sentences, words, characters, and character attributes along with page position and graphical images. Database access for these applications is typically a directed graph structure rather than the kind of ad hoc query that can be supported in SQL. Each object contains within its description reference to many other objects and elements. These are automatically collected by the object manager to provide the total view. In typical SQL applications, the developer makes explicit requests for related information.

In trying to manipulate such complex data using a relational system, a programmer writes code to map extremely complex in-memory data structures onto lower-level relational structures using awkward and resource-intensive recursive programming techniques. The programmer finds himself or herself doing database management instead of letting the DBMS handle it. Worse, even if the programmer manages to code the translation from in-memory objects to relational tables, performance is unacceptable.

Thus, relational systems have not been any help for the programmer faced with these complex coding tasks. The object-oriented programming paradigm, on the other hand, has proven extremely useful. The complex data structures CAD and CASE programmers deal with in memory are often defined in terms of C++ or Smalltalk objects.

It would be helpful if the programmer didn't have to worry about managing these objects, moving them from memory to disk, then back again when they're needed later. Some OOP systems provide this object "persistence" just by storing the memory image of objects to disk. But that solution only works for single-user applications. It doesn't deal with the important concerns of multiuser access, integrity, and associative recall.

Persistence means that objects remain available from session to session. Reliable means automatic recovery in case of hardware or software failures. Sharable means that several users should be able to access the data. All of these qualities may require systems that are larger than many that are currently available. In some cases, of course, programmers aren't dealing with overwhelmingly complex data, yet want to combine the increased productivity of object-oriented programming with the flexibility of an SQL DBMS. Relational technology has been extended to support binary large objects (BLOBs), text, image and compound documents, sound, video, graphics, animation, and abstract data types. As a result, organizations will be able to streamline paper-intensive operations to increase productivity and decrease business costs-assuming they use a database as a repository and manager for this data.

---7 ---

Client/Server Systems Development

---Hardware---

7.1 Executive Summary

"Advantages of Client/Server Computing," the cost of powerful hardware for client/server computing has declined dramatically in the last few years. Nevertheless, this power must be packaged properly, and cost still must be considered in the design and purchasing decision. Hardware that provides the client/server, LAN-to-LAN, and LAN-to-WAN connectivity must be acquired for clients, servers, data storage, and the networks.

Entry-level client workstations can range from a basic Intel-based PC to an entry-level Apple Macintosh or an X-Terminal. These entry-level clients start at about $1,000 and use LAN servers for printing, backup, software storage, application execution, and WAN connectivity. High-end client workstations can cost more than $50,000 for engineering stations that provide advanced capabilities such as a gigabyte or more of local storage, high-resolution graphics monitors, 100-MIPS processing, direct WAN connectivity, 1000-dpi color printing, or professional multimedia development tools. The average client workstation has dropped from $5000 to $2000 in the last two years. This buys a configuration with the processing power equivalent to an 8Mbyte Intel 33-MHz 486DX PC with local storage of 250Mbytes, LAN connectivity, and a VGA-equivalent monitor. This cost level is not expected to decline much further, because GUI software and reengineered application requirements will steadily increase the processing power requirements for entry-level machines.

Server hardware offers the largest and most complex set of choices. Servers run the gamut from a $30M+ traditional IBM mainframe, to a 4- to 16-way symmetric segment multiprocessor machine, to a 32- to 32767-processor massively parallel cluster supporting hundreds of users, to a $5,000 PC used to provide file and connectivity services for a small LAN workgroup. Many organizations also have client/server applications that use the services of existing IBM 370 mainframes running VM, MVS, or VSE, DEC VAX minicomputers running VMS or Ultrix, and large RISC-based systems running UNIX-all as high-end servers.

Other mainframe and minicomputer hardware platforms, running proprietary operating systems, are frequently used in terminal emulation mode from the client workstation. The non-IBM and DEC proprietary operating system platforms rarely are used to provide other services, such as database and RPC-invoked application services. There is a lack of tools available in these environments to build or buy client/server applications. Servers based on the IBM, DEC, and UNIX operating systems will provide application services using existing applications through terminal emulation or RPC-invoked application services. These same servers will provide connectivity and database services to the first client/server applications in an organization.

Connectivity requires every client workstation to be connected to a LAN or through a WAN to a remote server. In the usual situation, the workstation is connected through an Ethernet, Token Ring, FDDI, CODI, or occasionally a parallel or serial interface to the LAN. The primary connection types require a network interface card (NIC) to be inserted in the workstation to provide the protocol processing necessary to establish and maintain the connection. The cost of LAN connectivity has declined rapidly in parallel with the industry reduction in workstation costs.

Cabling costs vary widely, depending on the physical difficulty of installation and whether the network planners choose unshielded twisted-pair (UTP), shielded twisted-pair (STP), or glass-fiber cables. Cable costs without installation run from $1 per foot for UTP, $1.50 per foot for STP, to $3 per foot for glass fiber. Installation costs vary from $1 per foot to $15 per foot, depending on the physical environment and connection requirements. Glass-fiber termination equipment is more costly than twisted-pair, although the costs are declining. Current costs are between $100-200 for Ethernet, $300-500 for Token Ring, $300-700 for CDDI, and $750-1250 for FDDI.

Today, many vendors provide the hardware for these connections. Each vendor offers some advantages in terms of cost, performance, and reliability. Motorola provides wireless Ethernet connectivity at lower speeds and higher costs than wired connections. Wireless connections are an advantage in existing buildings with no cable installed and with relatively low-speed communications requirements.

WAN connectivity requires each workstation to be directly connected to the WAN or to a communications server connected to the WAN. Most new LANs are installed using communications servers. There are cost, performance, and especially network management reasons for using a LAN communications server. A substantial advantage accrues because there is no need to cable each workstation to the WAN. Workstations that are individually connected to the WAN require an embedded controller card for synchronous communications and either a modem or serial connection for asynchronous communications. These typically operate at speeds of 2400-64000 bits per second (bps) through analog or digital modems. Each workstation must have its own cable connecting it to the WAN controller. Workstations connected to the WAN through a communications server share a higher-speed connection, typically 14400 bps, 56000 bps, or 1.54 Mbps.

A major advantage of the communications server is its ability to multiplex a high-speed communications line and provide bandwidth on demand to each client workstation. Only the single LAN cable and LAN controller are needed to provide workstation connectivity in the server implementation.

Data storage can be provided to a client from a local disk or through the file services of the NOS. Local disk storage requires the workstation to have its own disk devices. Server storage involves large shared server disks. In either case, a backup capability must be provided. This can be done through local diskette or tape devices or though a server tape, disk, or optical device.

7.2 Hardware/N etwork Acquisition

Before selecting client hardware for end users, most organizations should define standards for classes of users. This set of standards simplifies the selection of the appropriate client hardware for a user and allows buyers to arrange purchasing agreements to gain volume pricing discounts.

There are a number of issues to consider when selecting the client workstation, including processor type, coprocessor capability, internal bus structure, size of the base unit, and so on. However, of these issues, one of the most overlooked regarding client/server applications is the use of a Gill. Gill applications require VGA or better screen drivers. Screens, larger than the 15-inch PC standard, are required for users who normally display several active windows at one time; the more windows active on-screen, the larger the monitor viewing area requirements. The use of image, graphics, or full-motion video requires a large screen with very high resolution for regular usage. It is important to remember that productivity is dramatically affected by inability to easily read the screen all day. Inappropriate resolution will lead to fatigue and inefficiency.

The enterprise on the desk requires that adequate bandwidth be available to provide responsiveness to the desktop user. If regular access to off LAN data is required, a router based internetworking implementation will be required. If only occasional off LAN access is required, bridges can be used. Routers provide the additional advantage of support for multiprotocol internetworking. This is frequently necessary as organizations install lOBaseT Ethernet into an existing Token Ring environment. Fast Ethernet and FDDI are becoming more prevalent as multimedia applications are delivered.

7.3 PC-Level Processing Units

Client/server applications vary considerably in their client processıng requirements and their 1/0 demands on the client processor and server. In general, clients that support protected-mode addressing should be purchased. This implies the use of 32-bit processors-perhaps with a 16-bit I/O bus if the I/O requirement is low. Low means the client isn't required to send and receive large amounts of data, such as images, which could be 1OOK bytes or larger, on a constant basis.

As multiwindowed and multimedia applications become prevalent during 1994, many applications will require the bandwidth only provided by a 32-bit I/O bus using VESA VL-bus or Intel PCI technology. Windowed applications require considerable processing power to provide adequate response levels. The introduction of application integration via DCE, OLE, and DOE significantly increases the process ing requirements at the desktop. The recommended minimum configuration for desktop processors has the processing capacity of a 33Mhz Intel 486SX. By early 1995, the minimum requirement will be the processing capacity of a 50Mhz Intel 486DX or a 33Mhz Intel Pentium.

7.4 Macintosh

The Mac System 7 operating system is visually intuitive and provides the best productivity when response time to GUI operations is secondary. The Motorola 68040, 8Mbytes RAM, 120Mbyte disk is recommended. By early 1995, the availability of PowerPC technology and the integration of System 7 with AIX and Windows means that users will need considerably more processor capacity. Fortunately, the PowerPC will provide this for the same or lower cost than the existing Motorola technology.

7.5 Notebooks

Users working remotely on a regular basis may find that a notebook computer best satisfies their requirements. The notebook computer is the fastest growing market today. The current technology in this area is available for Intel PC, Apple Macintosh, and SPARC UNIX processors. Because notebooks are "miniaturized, " their disk drives are often not comparable to full-size desktop units. Thus, the relatively slower speed of disk I/O on notebooks makes it preferable to install extra RAM, creating "virtual" disk drives.

A minimal configuration is a processor with the equivalent processing power of a 33Mhz Intel 486SX, 8mbytes of RAM and 140Mbytes of disk. In addition, the notebook with battery should weigh less than seven pounds and have a battery life of three hours. Color support is an option during 1994 but will be mandatory for all by 1995. In addition, if the application will run a remote GUI, it is desirable to install software to compress the GUI and V.32 modem communications at 9600 bps or V.32bis at 14400 bps, employing V.42 and V.42bis compression, respectively. The effective throughput is two to three times the baud rate because of compression. The use of MNP4 and V.42 or MNP5 and V.42bis error correction enables these speeds to work effectively even during noisy line conditions. The introduction of PCM CIA technology, credit card size modems, and flash memory are available to upgrade the notebook.

7.6 Pen

Pen-based clients provide the capability to operate applications using a pen to point and select or write without the need for a mouse or keyboard. Frequently, they are used for verification, selection, and inquiry applications where selection lists are available. Developers using this technology use object-oriented software techniques that are RAM-intensive.

The introduction of personal digital assistant (PDA) technology in 1993 has opened the market to pocket size computing. During 1994, this technology will mature with increased storage capacity through cheaper, denser RAM and flash memory technology. The screen resolution will improve, and applications will be developed that are not dependent upon cursive writing recognition.

The PDA market is price-sensitive to a $500-$1000 device with the capability to run a Windows-like operating environment in 4MB of RAM, a 2QMhz Intel 486SX processor, and 8MB of flash memory. Devices with this capability will appear in 1994, and significant applications beyond personal diaries will be in use. During 1995, 16MB of RAM and 32MB of flash memory will begin to appear, enabling these devices to reach a mass market beyond 1996. In combination with wireless technology advances, this will become the personal information source for electronic news, magazines, books, and so on. Your electronic Personal Wall Street Journal will follow you for access on yourPDA.

7.7 UNIX Workstation

UNIX client workstations are used when the client processing needs are intensive. In many applications requiring UNIX, X-terminals connected to a UNIX presentation server will be the clients of choice. A UNIX client workstation will then have more processing power than a PC client.

The introduction of software from SunSoft, Insignia Solutions, and Locus Computing that supports the execution of DOS and Windows 3.x applications in a UNIX window makes the UNIX desktop available to users requiring software from both environments. The PowerPC and Spare technologies will battle for this marketplace. Both are expected to gain market share from Intel during and after 1994.

7.8 X-Terminals

X-terminals provide the capability to perform only presentation services at the workstation. Processing services are provided by another UNIX, Windows 3.x, NT, OS/2 2.x, or VMS server. Database, communications, and applications services are provided by the same or other servers in the network. The minimum memory configuration requirement for an X-terminal used in a client/server application is 4-8 Mbytes RAM, depending on the number of open windows.

7.9 Server Hardware

Server requirements vary according to the complexity of the application and the distribution of work. Because servers are multiuser devices, the number of active users is also a major sizing factor. Servers that provide for 32-bit preemptive multitasking operating systems with storage protection are preferred in the multiuser environment.

Intel-based tower PCs and Symmetric Multi-Processors (SMPs) are commonly used for workgroup LANs with file and application service requirements. Most PC vendors provide a 66Mhz Intel 486DX or Intel Pentium for this market in 1994. SMP products are provided by vendors such as IBM, Compaq, and NetFrame. Traditional UNIX vendors, such as Sun, HP, IBM, and Pyramid provide server hardware for applications requiring UNIX stability and capacity for database and application servers and large workgroup file services.

The SMP products, in conjunction with RAID disk technology, can be configured to provide mainframe level reliability for client/server applications. It is critical that the server be architected as part of the systems management support strategy to achieve this reliability.

7.10 Data Storage

Permanent storage requirements are very application-specific. In addition to quantity of disk storage, the issues of performance and reliability must be considered.

7.11 Magnetic Disk

Disk storage devices should use the SCSI-2 standard controller interface. This provides the best performance in a standards-based environment. Many vendors provide high-capacity, high-performance, and highly reliable disk devices for this controller.

The use of high-capacity cache storage dramatically improves performance. Most current SCSI-2 controllers are configurable with 256K or more cache. This is an important, yet frequently overlooked component of the architecture. New .drives are available in the traditional 3.5 size with l.O-l.6Gbyte capacity. The use of compression software can easily double this capacity. With the increasing size of GUI software and the introduction of multimedia applications, the demand for disk capacity will increase rapidly during 1994 and beyond.

7.12 Mirrored Disk

When applications require high reliability, it may be appropriate to use a configuration that supports mirrored disks. With this configuration, data is automatically written to two disks. This enables the application to continue if a failure occur on one disk. System files and data files should be considered for mirroring. Even though system files are usually read-only, the number of users affected by unavailability of the files may justify this redundancy. In addition, performance can improve since dual reads can be handled in parallel.

7.13 RAID-Disk Array

Traditional magnetic disk technology is often referred to as single large expensive disk (SLED). Very high performance and high availability can be achieved through a redundant array of inexpensive drives (RAID). These enable data files to be spread across several physical drives. Data also can be mirrored as part of this configuration. RAID technology provides a considerable performance advantage because many parallel I/O operations can be processed at the same time. High capacity caches must be used in conjunction with RAID to achieve optimum performance. The size will be identified as part of the architecture definition.

7.14 Tape

Although most permanently stored data uses disk, tape is a very popular form of low-cost magnetic storage and is used primarily for backup purposes. The standard backup tape device today is digital audio tape (DAT). These tapes provide approximately 1.2 Gbytes of storage on a standard cartridge-size cassette tape. Tape is a sequential medium and does not adequately support direct (random) access to information. If an organization standardizes on a single tape format and technology, distribution of information by mailing tapes can be a cost-effective communications mechanism for large quantities of information that do not require real-time transmission or accessibility.

7.15 Optical Disks

Optical disk storage technology provides the advantage of high-volume, economical storage with somewhat slower access times than traditional magnetic disk storage.

7.16 CD-ROM

Compact disk-read only memory (CD-ROM) optical drives are used for storage of information that is distributed for read-only use. A single CD-ROM can hold up to 800MB of information. Software and large reports distributed to a large number of users are good candidates for this media. CD-ROM also is more reliable for shipping and distribution than magnetic disks or tapes.

By 1995, it is expected that all software and documentation will be distributed only on CD-ROM. The advent of multimedia applications and the resulting storage requirements will further drive the demand for CD-ROM.

In 1993, the speed of CD-ROM technology was doubled through a doubling of the rotation of the drive. Newer drives will triple-spin and quad-spin. The speed of the drive is very critical for applications that use the CD-ROM interactively. The addition of large cache SCSI-2 controllers can also significantly improve performance. The architecture definition must look at the business requirement in determining the appropriate configuration. Poor selection will result in unacceptable performance, excessive cost, or both.

7.17WORM

Write once, read many (WORM) optical drives are used to store information that is to be written to disk just once but read many times. This type of storage is frequently used to archive data that should not be modified. Traffic tickets issued by police departments are scanned and stored on WORM drives for reference on payment or nonpayment. The WORM technology guarantees that the image cannot be tampered with. A magnetic drive can be used to store an index into the data on the WORM drive. Data can be effectively erased from a WORM by removing reference to it from the index. This can be useful when a permanent audit trail of changes is required.

7.18 Erasable Optical

Erasable optical drives are used as an alternative to standard magnetic disk drives when speed of access is not important and the volume of data stored is large. They can be used for image, multimedia, backup, or high-volume, low-activity storage.

7.19 Network Interface Cards (NICs)

Client and server processors are attached to the LAN through NICs. These provide the physical connectivity to the wire and the protocol support to send/receive messages. As discussed in Chapter 5, "Components of Client/Server Applications-Connectivity," the most popular network protocols today are Token Ring, Ethernet, and FDDI. The following paragraphs illustrate key selection issues regarding >each technology.

7.20 Token Ring

Token Ring NICs were originally IBM-only products but are now provided and supported by many PC and UNIX vendors. The IEEE standard 802.5 defines the standards for the interface. Token Ring NICs are particularly desirable for LANs that are collocated with an IBM mainframe. They are also useful when interactive use is combined on the same LAN with high-volume file transfer or print image communications. Token Ring LANs operate at 4 or 16 Mbps. Shielded twisted-pair (STP) (Type I cabling) is required by some vendors for Iô-Mbps processing, but unshielded twisted-pair (UTP) cable is supported by many at 16 Mbps and all at 4 Mbps.

The rapid decline in price for lOBaseTEthernet and the increasing availability of Fast Ethernet means that despite some technical advantages the future of Token Ring is limited.

7.21 Ethernet

The existing de facto standard for LAN connection defined by the IEEE standard 802.3, Ethernet is supported by almost every vendor. The large number of vendors providing NICs ensures their competitive pricing. Ethernet works well when interactive-only or file transfer-only communications are present on the LAN. When mixing interactive and file transfer on the same Ethernet system, performance is excellent when LAN loading does not exceed 30 percent of the capacity. Most Ethernet LANs operate at 10 Mbps. The present standard for Ethernet connectivity IOBaseToperates on STP or UTP. Recent products supporting Fast Ethernet and ATM will provide support for lOOMbit and up to 2.4Gbit on existing Type 5 DTP-cabled network.

7.22 FDDI

Fiber Distributed Data Interchange (FDDI) is a protocol originally defined for high-speed communications over glass fiber. FDDI provides 100-Mbps throughput today. NICs for FDDI are becoming available for more processing environments. This throughput is necessary when applications deal with large images, large file transfers, or multimedia using full-motion video. The rapid advances in Fast Ethernet and ATM means that FDDI will see limited rollout except for building internetworking and WANs.

7.23 CDDI

Copper Distributed Data Interchange (CODI) provides support for FDDI communications over copper wire. The same 100-Mbps throughput is supported over Type 1 cabling (STP), and standards are emerging to provide support over Type 3 cabling (UTP) that is carefully selected and installed.

7.24 Power Protection Devices

A lot has been written in books, magazines, and journals about computer hardware and software; and a number of computer specialty businesses are dedicated to helping you work through issues of specific concern to your business objectives. Rather than cover the minute details here, this chapter has attempted to highlight a number of areas for you.

However, before closing this chapter, one critical area often overlooked (but is the cause of many serious problems when neglected) is power protection.

7.25 Uninterruptible Power Supply (UPS)

Prices for UPS have declined to the point where they are widely used for LAN server protection. These units contain battery backups to enable at least a graceful power-down sequence. All buffers can be purged and files closed so that no information is lost. Other units provide 15-90 minutes of power backup to handle most power failure situations.

7.26 Surge Protectors

The electronics in computer systems are affected by power fluctuations. Protection can be obtained through the use of surge protection equipment. Every client and server processor, and all peripherals should be wired through a surge protector. Most UPS systems include integrated surge protectors.

CONCLUSION

Now that we've discussed some of the major components of networks and TCP/IP, you have the necessary background to examine the more critical issues of security in a converged environment. Knowing how networks are built gives you a better understanding of what physical or logical vulnerabilities are introduced by choosing one particula,r network design over another. Knowing how packets are formed gives you- a better understanding of how they can be crafted or modified to achieve a specific purpose. Knowing how packets are transmitted and delivered gives you a better understandiig of what can happen to packets as they travel from source to destination. A good understanding of the basics of networking and TCP/IP is critical to identifying, understanding, and correcting vulnerabilitiesin your converged-environmem,

We know that briefly, Network clients request information or a service from a server, and that server responds to the client by acting on that request and returning results. This approach to networking has proven to be a cost-effective way to share data between tens or hundreds of clients. Usually the client and server are two separate devices on a LAN, but client/server systems work equally well on long-distance WAN's (including the Internet).

FIGURES & TABLES

NetW'are Architecture

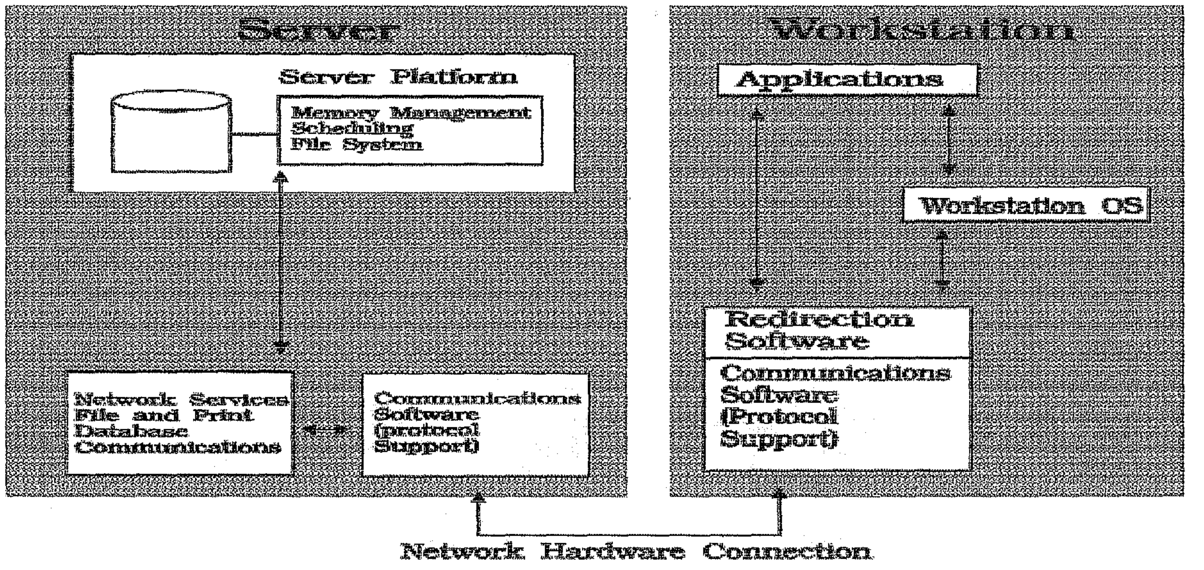

Figure -1- Network Hardware

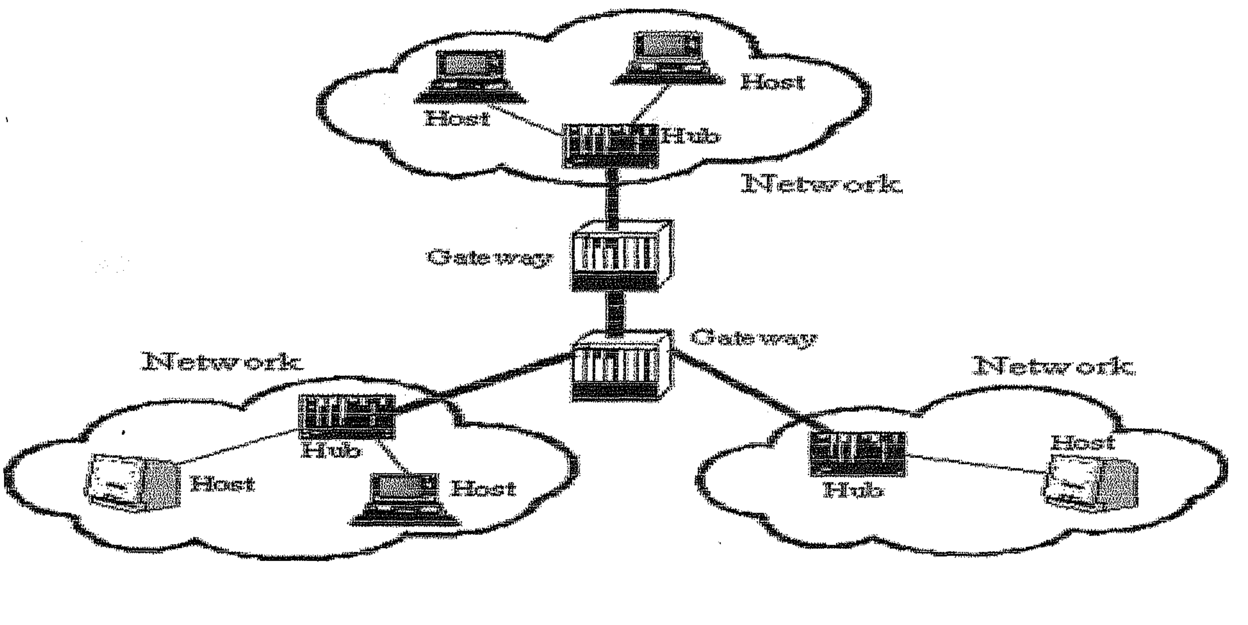

Figure -2- DNS Transaction through a Router

Netmask.	11111111	11111111	11111111	0000	0000
IPAddress	10000000	11000000	00001010	10100101	

Network Address — Host Address

Figure -3- Netmask

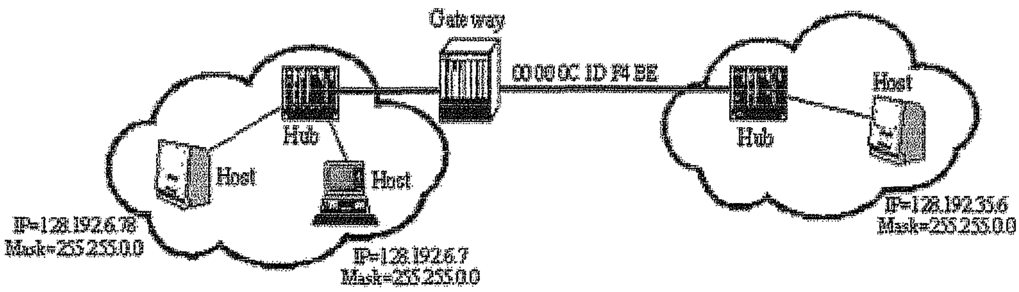

Figure -4- Proxy Arp

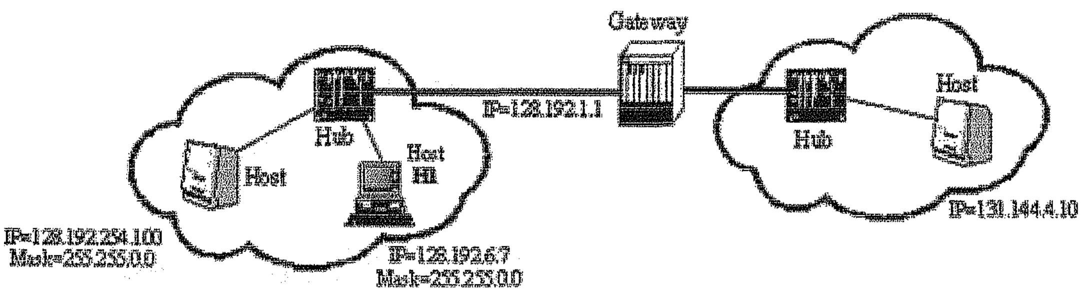

Figure -5- Host Routing

entry	netmask	net-address	next-hop	hop-count	**(comments)**
1	255.255.0.0	128.192.0.0	128.192.6.7	0	DCN
2	0.0.0.0	0.0.0.0	128.192.1.1	1	DR

Table -1- Host Routing

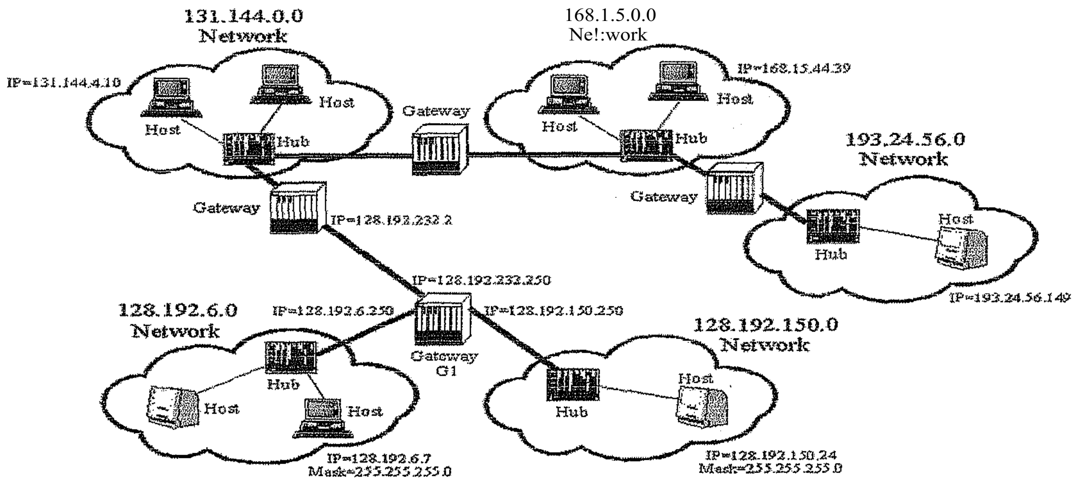

Figure -6- Gateway Routing

entry	netmask	net-address	next-hop	hop-[count]	(comments)
1	255.255.255.0	128.192.6'.0	128.193.6.250	0	DCN
2	255.255.255.0	128.192.7.0	128.192.7.250	0	DCN
3	255.255.255.0	128.192.150.0	128.192.150.250	0	DCN
4	255.255.255.0	128.192.232.0	128.192.232.250	0	DCN
5	255.255.255.255	13.1.144.4.10	128.192.232.2	1	IfSR
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
7	0.0.0.0	0.0.0.0	n2s.192.232.2	1	DR

Table -2- Gateway Routing

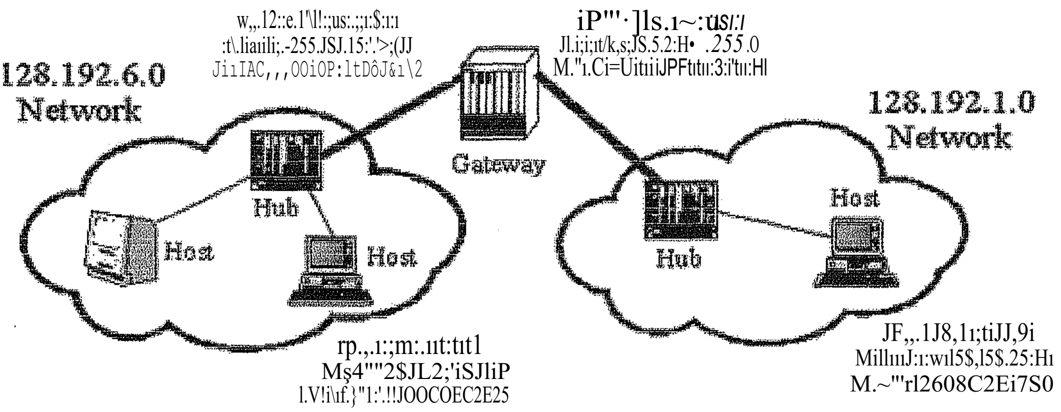

Figure -7- Network Topology

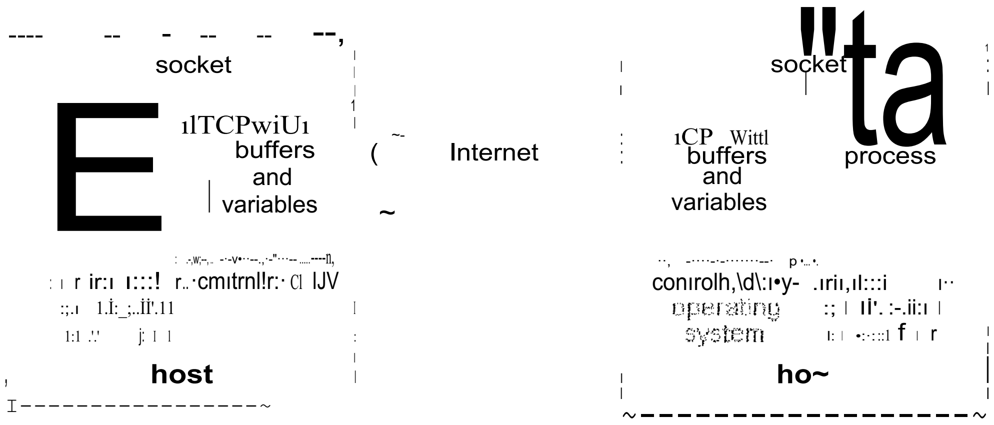

Figure -8- Tep Socket

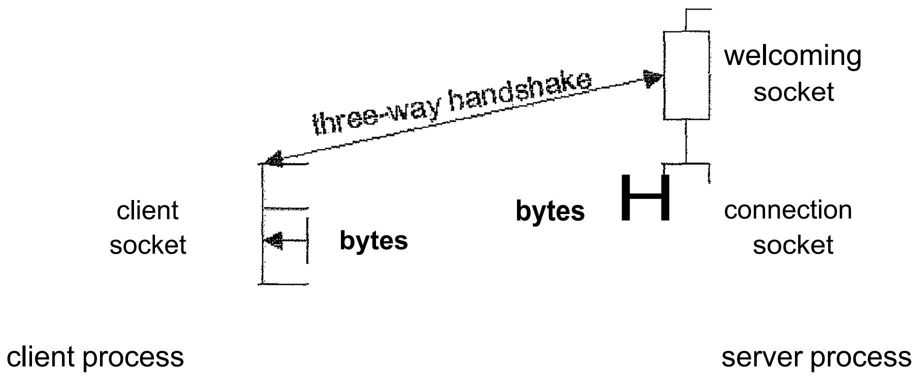

Figure -9- Client Socket

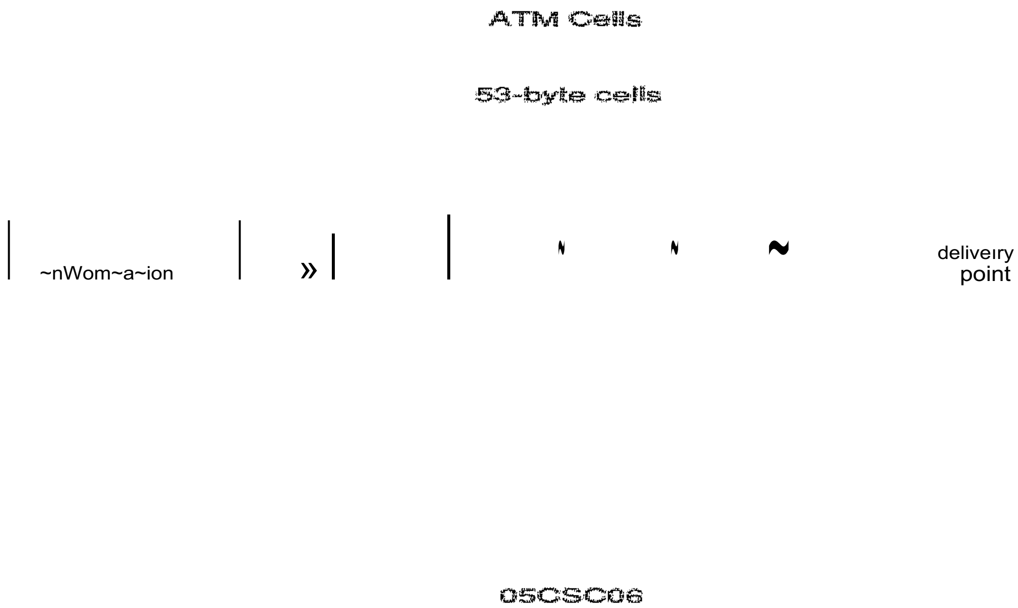

Figure -10- ATM Cells

REFERENCES

1) http://wvvw.officewizard.com/books/clientserver/ClientServerComputing.htm

2) http://freebooks.by.ru/view /ClientServerComputing/index.htm

3) Robert Orfali and Dan Harkey, Client-Server Programming with OS/2 (New York: Van Nostrand Reinhold, 1991)

4) Amdahl Corporation, Globalization, The IT Challenge (Amdahl Corporation, 1950)

5) Michael L. Dertouzos, "Building the Information Marketplace," Technology Review, No. 94, (January 1991)

6) Edelstein, Herbert A., "Database World Targets Next-Generation Problems," Software Magazine Vol. VII, No. 6 (May 1991)

7) Jerry Cashin, "OSI DEC Attempt to Add OSI Service," *Software Magazine* 11, No. 3 (March 1991)

8) I'Reinventing Companies," The Economist 321, NO. 7728 (October 12, 1991)

9) J.W. Semich, "The Distributed Connection:DEC," Datamation 37, No. 15 (August 1, 1991)

10) Network World 8, No. 40 (October 7, 1991)