NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

CLIENT/SERVER COMPUTING

Graduation Project
COM- 400

Student: Ayman Al-Amouri (991578)

Supervisor: Assoc. Prof. Dr Dogan Ibrahim

Nicosia - 2002

NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

CLIENT/SERVER COMPUTING

Graduation Project
COM- 400

Student: Ayman Al-Amouri (991578)

Supervisor: Assoc. Prof. Dr Dogan Ibrahim

Nicosia - 2002

ACKNOWLEDGMENT

"First I would like to express my thanks to my supervisor Assoc. Prof. Dr Dogan Ibrahim for supervising my work. Under the guidance of him I successfully overcame many difficulties and learned a lot about Client/Server. In each discussion, he used to explain the problems and answer my questions. While teaching, he always helped me a lot, in my study or in my life, and I felt remarkable progress from his advisorship.

On this note I would like to express my gratitude to the Near East University for the awareness that they granted to us. I would also like to thank my teachers who supported me and taught me the true meaning of determination, I am forever in their gratitude for having the belief in me and for not giving up on me.

Special thanks for my parents and all my family for moral and material support, and for helping me to complete my study and to become an engineer.

Finally, I want to thank all my friends who support me all the time and for spent a nice days with them in this university."

ABSTRACT

What is the nature of the process of implementing a new technology? How should the dynamics of implementing a new technology be studied? What research methods are best-suited to the study of complex issues of social and organizational impacts arising from the implementation of a new technology? Client-server computing represents a significant new technology that has not been a focus of research investigations. As companies pursue client-server technology as a replacement for legacy computing systems, there is a need to provide practitioners with grounded research that discover patterns of organizational and social dynamics that influence the successful outcome of a transition to this new technology. This article suggests that naturalistic research studies can formulate realistic business foundations for the successful implementation of client-server computing.

Tremendous growth of client/server (C/S) computing is expected throughout the 1990s. Computer information systems (CIS) curricula are faced with the ongoing challenge of providing client-server instruction while technologies and methodologies are still evolving. To keep pace with this dynamic environment, faculty must begin building initial client/server experiences into the CIS curriculum.

The purpose of this thesis is to describe the fundamental concepts of client/server. next, the advantages of the client-server computing that made this module to be cosidered from the most important organizations in the world are listed. Finally, future client/server changes are considered.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

IT	Information Technology
IS	Information Systems
OS	Operating Systems
GUI	Graphical User Interface
SDE	Systems Development Environment
SQL	Structured Query Language
WAN	Wide Area Network
BLOBs	Binary Large Objects
LAN	Local Area Network
PC	Personal Computer
OLE	Object Linking and Embedding
OLTP	Online Transaction Processing
NOS	Network Operating System
CORBA	Communicating Object Request Broker Architecture
EHLLAPI	Extended High Level Language Application Program Interface
IPC	Interprocess Communication
RPC	Remote Procedure Call
NFS	Network File System
BIOS	Basic Input/Output System
IPL	Initial Program Load
E-PROM	Erasable Programmable Read-Only Memory
NIC	Network Interface Card
OLE	Object Linking and Embedding
SDK	Software Development Kit
OMG	Object Management Group
ENS	Enterprise Network Services
SAA	System Application Architecture
OPT	Open Protocol Technology
ODI	Open Datalink Interface
NLM	NetWare Loadable Module

NT	New Technology
CUA	Common User Access
CPI	Common Programming Interfaces
ER	Entity Relationship
IEEE	Institute of Electrical and Electronic Engineers
UTP	Unshielded Twisted Pair
STP	shielded twisted pair
SNMP	Simple Network Management Protocol
CMIP	Common Management Information Protocol
OSI	Open Systems Interconnect
MAC	Media Access Control
LLC	Logical Link Control
ATM	Asynchronous Transfer Mode
ISDN	Integrated Services Digital Network
B-ISDN	Broadband Integrated Services Digital Network
TCP/IP	Transmission Control Protocol/Internet Protocol
DARPA	Defense Advanced Research Projects Agency
ARP	Address Resolution Protocols
FTP	File Transfer Protocol
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
APPC	Application Program-to-Program Communication
TQM	Total Quality Management
CASE	Computer-Aided Software Development
DBMS	Data-Base Management System
OODBMS	Object-Oriented Database Management System
ADDS	Application Driven Database Systems
BPR	Business Process Reengineering
UPS	Uninterruptible Power Supply
DME	Distributed Management Environment
DCE	Distributed Computing Environments
MIS	management information systems
SPA	Software Publishers' Association
OURS	Open User Recommended Solutions

LMF	License Management Facility
SRs	Service Requests
DCS	Distributed Computing System
ARPANET	Advanced Research Projects Agency Network
DOD	Department of Defense
OO	Object Orientation
OOUI	Object Oriented User Interfaces
OMG's	Object Management Group's
ORB	Object Request Broker
TOD	Time of Day
API	Application Programming Interface
OMG	Object Management Group's
OCSI	Object-Oriented Client/Server Internet

Introduction

Along with the tremendous potential, however, comes tremendous confusion and chaos in the marketplace. Open systems, object orientation, graphical user interfaces, UNIX, OS/2, CASE, database, and superservers—these are terms that can impact information systems choices in various ways. But in today's rapidly changing business and computing environments, how do we decide which solution is best for our needs? And how do we go about implementing that solution?.

The Client/Server Computing provide answers to these and similar questions. The Information in Client/Server Computing comes from years of experience and first-hand implementation of new technologies.

In a competitive world it is necessary for organizations to take advantage of every opportunity to reduce cost, improve quality, and provide service. Most organizations today recognize the need to be market driven, to be competitive, and to demonstrate added value.

A strategy being adopted by many organizations is to flatten the management hierarchy. With the elimination of layers of middle management, the remaining individuals must be empowered to make the strategy successful. Information to support rational decision making must be made available to these individuals. Information technology (IT) is an effective vehicle to support the implementation of this strategy; frequently it is not used effectively. The client/server model provides power to the desktop, with information available to support the decision-making process and enable decision-making authority.

Client/server computing is a phrase used to describe a model for computer networking. This model offers an efficient way to provide information and services to many users.

A network connection is only made when information needs to be accessed by a user. This lack of a continuous network connection provides network efficiency.

In client/server computing, processes are divided between the client and the server. This relationship is based on a series of requests and responses.

- ***Client:*** Requests services or information from another computer (the server computer).

- ***Server:*** Responds to the client's request by sending the results of the request back to the client computer.

In a client/server setting, the client computer runs a software application called a client program. This software allows a computer to act as a client.

The client program:

- Enables the user to send a request for information to the server.

- Formats the request so that the server can understand it.

- Formats the response from the server in a way that the user can read.

In a client/server setting, the server computer runs a software application called a server program. This software allows a computer to act as a server.

The server program:

- Receives a request from a client and processes the request.

- Responds by sending the requested information back to the client.

Most transactions that occur on the Internet are client/server based. Some examples include:

- FTP (file transfer protocol) - An FTP client program contacts an FTP server and requests the transfer of a file; the FTP server responds by transferring the file to the client.

- WWW (World Wide Web) - In this case the client program is a browser. A browser requests the contents of a web page and displays the results on the user's computer.

- E-MAIL - A mail client program enables the user to interact with a server in order to access, read and send electronic mail messages.

In Summary we can say Client/server computing is a common networking model which enables many users to access information in an efficient manner. Generally, the user's computer is called the client and the machine that contains the information being accessed is called the server.

The client computer runs an application called a client program. A client program enables a user to send a request for information to the server and read the results that the server sends back. The server computer runs a server program which processes requests and sends results back to the client. Most Internet transactions, such as FTP, e-mail and accessing web pages are based on client/server networking.

The objective of this project is to prove that the client/server technology capable of bridging the chasm between user expectation and the ability of inforation system (IS) organization to fulfill this expectation.

The firt chapter represents an overview about the client/server computing architecture and its components, and talk about its evolution, application modules, features and characterstic of the client/server module.

Chapter two is devoted to the advantages of the client/server computig modules, and also the strong need for the various organizations to take care about this module.

Chapter three concerned to the basic fundementals of the client/server computing, Chapter provides the internet and the world wide web components and show how they tie with each other.

Chapters four, five and six are devoted to talk about the components of the client/server computing and its applications that are the client, server and connectivity respectively. Each chapter studies different component of the client/server computing and provides its services, applications, rules, implementation, and the way of connecting with the other components in order to achive the maximum benefits.

CHAPTER ONE

OVERVIEW OF CLIENT/SERVER COMPUTING

1.1General

Client/server is a computational architecture that involves client processes requesting service from server processes. Client/server computing is the logical extension of modular programming. Modular programming has as its fundamental assumption that separation of a large piece of software into its constituent parts ("modules") creates the possibility for easier development and better maintainability.

Client/server computing takes this a step farther by recognizing that those modules need not all be executed within the same memory space. With this architecture, the calling module becomes the "client" (that which requests a service), and the called module becomes the "server" (that which provides the service). The logical extension of this is to have clients and servers running on the appropriate hardware and software platforms for their functions. For example, database management system servers running on platforms specially designed and configured to perform queries, or file servers running on platforms with special elements for managing files. It is this latter perspective that has created the widely-believed myth that client/server has something to do with PCs or Unix machines.

1.2 Evolution of Client-Server Computing

The evolution of Client-Server Computin has been driven by business needs, as well as the increasing costs for host (mainframe and midrange) machines and maintenance, the decreasing costs and increasing power of micro-computers and the increased reliability of LANs(Local Area Networks).

In the past twenty years, there are dramatic improvements in the hardware and software technologies for micro-computers. Micro-computers become affordable for small businesses and organisations. And at the same time their performances are becoming more and more reliable. On the other hand, the drop in price for mainframe is growing at a slower rate than the drop in its price. Little developments have achieved with mainframes.

The following are the improvements made by micro-computers:

- **Hardware:** The speed of desktop microprocessors has grown exponenetially, from a 8MHz 386-based computers to 100Hz-based pentium-based microprocessors. These mass-produced microprocessors are cheaper and more powerful than those used in mainframe and midrange computers. On the other hand, the capacity of main memory in micro-omputers has been quafrupling every three years. Typically main memory size is 16 Megabytes nowadays. Besides, the amount of backup storage and memory such as hard disks and CD-ROMs that are able to support micro-computers has also puts an almost unlimited amount of data in reach for end-users.
- **Software:** The development and acceptance of GUIs (Graphical User Interfaces) such as Windows 3.1 and OS/2 has made the PC working environment more user-friendly. And the user are more efficient in learning new application softwares in a graphical environment. Besides GUIs, the use of multithreaded processing and relational databases has also contributed to the popularity of Client-Server Computing.

1.3 Configurations in Client-Server Computing

Client-Server Computing is divided into three components, a Client Process requesting service and a Server Process providing the requested service, with a Middleware in between them for their interaction.

1.3.1 Client

A Client Machine usually manage the user-interface portion of the application, validate data entered by the user, dispatch requests to server programs. It is the front-end of the application that the user sees and interacts with. Besides, the Client Process also manages the local resources that the user interacts with such as the monitor, keyboard, workstation, CPU and other peripherals.

The client is a process (program) that sends a message to a server process (program), requesting that the server perform a task (service). Client programs usually manage the user-interface portion of the application, validate data entered by the user, dispatch requests to server programs, and sometimes execute business logic. The client-based process is the front- end of the application that the user sees and interacts with.

The client process contains solution-specific logic and provides the interface between the user and the rest of the application system. The client process also manages the local resources that the user interacts with such as the monitor, keyboard, workstation CPU and peripherals. One of the key elements of a client workstation is the graphical user interface (GUI). Normally a part of operating system i.e. the window manager detects user actions, manages the windows on the display and displays the data in the windows.

1.3.2 Server

On the other hand, the Server Machine fulfills the client request by performing the service requested. After the server receives requests from clients, it executes database retrieval , updates and manages data integrity and dispatches responses to client requests. The server-based process may run on another machine on the network; the server is then provided both file system services and application services.Or in some cases, another desktop machine provides the application services. The server acts as software engine that manages shared resources such as databases, printers, communication links, or high powered-processors. The main aim of the Server Process is to perform the back-end tasks that are common to similar applications.

The simplest form of servers are disk servers and file servers. With a file server, the client passes requests for files or file records over a network to the file server. This form of data service requires large bandwidth and can slow a network with many users. The more advanced form of servers are Database servers, Transaction server and Application servers.

The following diagram illustrates the relationship between client and server computers. The client requests information; the server processes the request and sends a response back to the client.

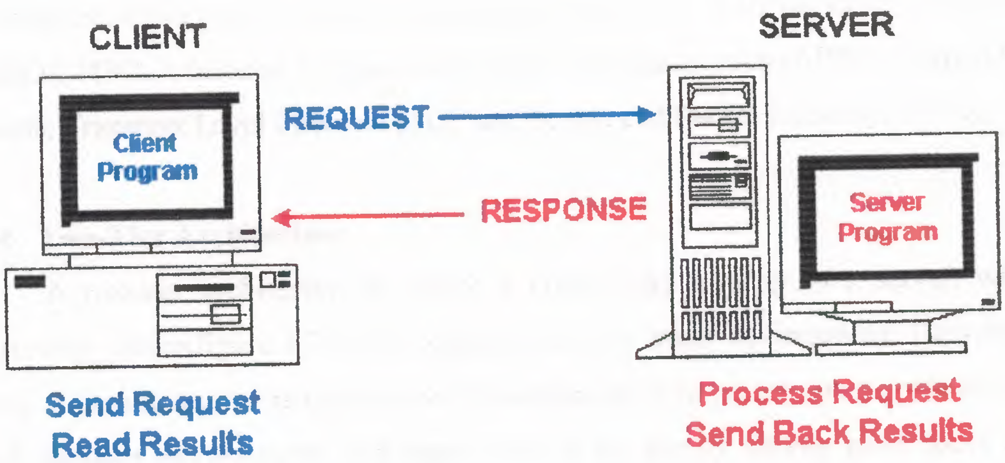

Figure 1.1. the relationship between client and server computers

1.3.3 Middleware

Middleware allows applications to transparently commnicate with other programs or processes regardless of location. The key element of Middleware is NOS (Network Operating System) that provides services such as routing, distribution, messaging and network management service. NOS rely on communiction protocols to provide specific services. Once the physical connection has been established and transport protocols chosen, a client-server protocol is required before the user can access the network services. A client-server protocol dictates the manner in which clients request information and services from a server and also how the server replies to that request.

The protocols are divided into three groups: media, transport and client-server protocols. Media protocols determine the type of physical connections used on a network (some examples of media protocols are Ethernet, Token Ring, Fiber Distributed Data Interface (FDDI), coaxial and twisted-pair). A transport protocol provides the mechanism to move packets of data from client to server (some examples of transport protocols are Novell's IPX/SPX, Apple's AppleTalk, Transmission Control Protocol/ Internet Protocol (TCP/IP), Open Systems Interconnection (OSI) and Government Open Systems Interconnection Profile(GOSIP)). Once the physical connection has been established and transport protocols chosen, a client-server protocol is required before the user can access the network services. A client-server protocol dictates the manner in which clients request information and services from a server and also how the server replies to that request (some examples of client- server protocols are NetBIOS, RPC, Advanced Program-to-Program Communication (APPC), Named Pipes, Sockets, Transport Level Interface (TLI) and Sequenced Packet Exchange (SPX)).

1.3.4 Two-Tier Architecture

A two-tier architecture is where a client talks directly to a server, with no intervening server(figure 1.2). It is typically used in small environments (less than 50 users). A common error in client/server development is to prototype an application in a small, two-tier environment, and then scale up by simply adding more users to the server. This approach will usually result in an ineffective system, as the server becomes overwhelmed. To properly scale to hundreds or thousands of users, it is usually necessary to move to a three-tier architecture.

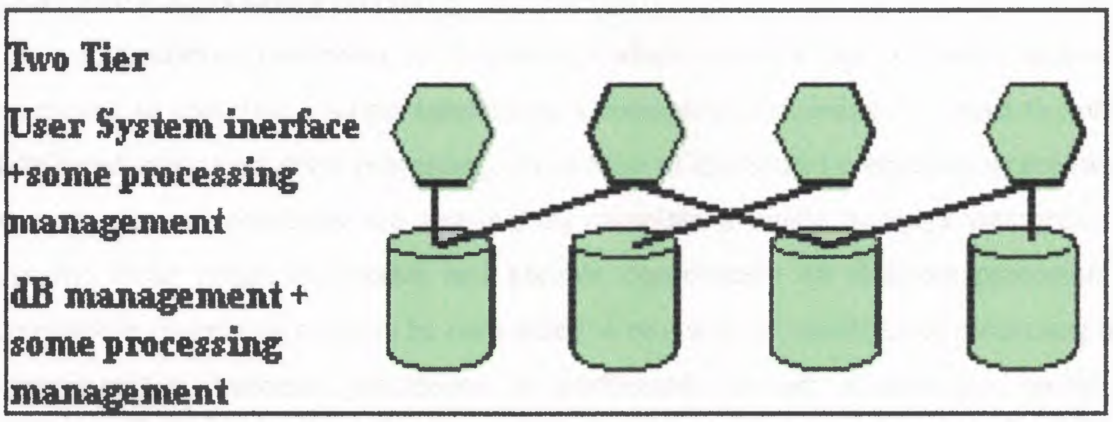

Figure 1.2. Two Tier Client Server Architecture Design

1.3.5 Three-Tier Architecture

A three-tier architecture introduces a server (or an "agent") between the client and the server. The role of the agent is manyfold. It can provide translation services (as in adapting a legacy application on a mainframe to a client/server environment), metering services (as in acting as a transaction monitor to limit the number of simultaneous requests to a given server), or intellegent agent services (as in mapping a request to a number of different servers, collating the results, and returning a single response to the client.

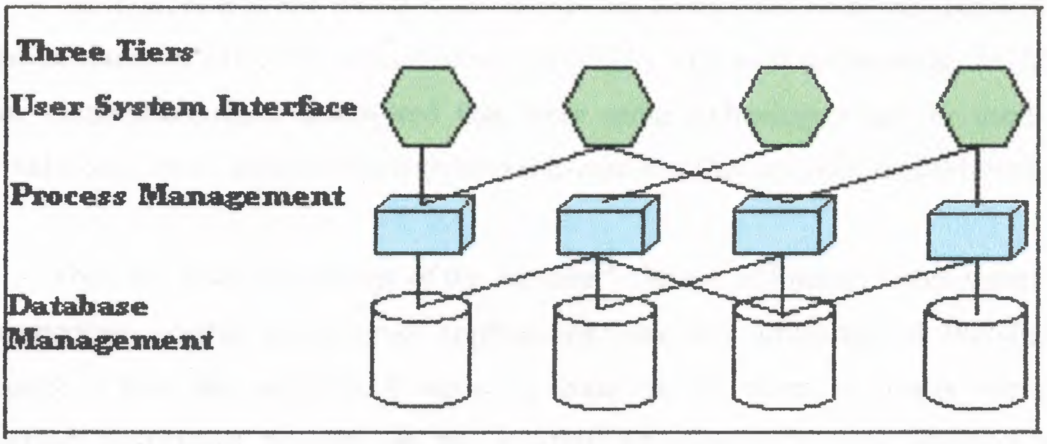

Figure 1.3. Three tier distributed client/server architecture depiction

1.3.6 The Cooperative Processing

Cooperative processing is computing which requires two or more distinct processors to complete a single transaction. Cooperative processing is related to both distributed and client/server processing. It is a form of distributed computing where two or more distinct processes are required to complete a single business transaction. Usually, these programs interact and execute concurrently on different processors. Cooperative processing can also be considered to be a style of client/server processing if communication between processors is performed through a message passing architecture.

1.3.7 The Distributed Processing

The distribution of applications and business logic across multiple processing platforms. Distributed processing implies that processing will occur on more than one processor in order for a transaction to be completed. In other words, processing is distributed across two or more machines and the processes are most likely not running at the same time, i.e. each process performs part of an application in a sequence. Often the data used in a distributed processing environment is also distributed across platforms.

1.3.8 The Intranet

The explosion of the World Wide Web is due to the world-wide acceptance of a common transport (TCP/IP), server standard (HTTP), and markup language (HTML). Many corporations have discovered that these same technologies can be used for internal client/server applications with the same ease that they are used on the Internet.

Thus was born the concept of the "Intranet": the use of Internet technologies for implementing internal client/server applications. One key advantage of Web-based intranets is that the problem of managing code on the client is greatly reduced. Assuming a standard browser on the desktop, all changes to user interface and functionality can be done by changing code on the HTTP server. Compare this with the cost of updating client code on 2,000 desktops. A second advantage is that if the corporation is already using the Internet, no additional code needs to be licensed or installed on client desktops. To the user, the internal and external information servers appear integrated. A rapidly-disappearing disadvantage is that there is limited ability to provide custom coding on the client. In the early days of the Web, there were limited ways of interacting with the client. The Web was essentially "read-only". With the release of code tools such as Java and JavaScript, this limitation is no longer a major issue.

1.4 The Four Dominant Client/Server Application Models

Having had a deeper look into the terms and architectures of client/server technology, let's consider the dominant application models available. Nowadays, there are four client/server application models that are widely used in the market. They are Structured Query Language (SQL) databases, Transaction Processing (TP) monitors, groupware and distributed objects. Each one of them is capabable of creating its own complete client/server applications with its own tools. Moreover, they also introduce their own favourable form of middleware (all this will be further discussed later). But first, what is the reason for having different models instead of having just one model, and what is the advanatges/disadvantages of having just one particular model.

The reason why we need different models for different applications is because each one of them have their own advantages and disadvantages, and sometimes one model performs better than the others in one particular situation. Furthermore, standardising the whole market with one particular model will not only discourage the vendors from developing other new (and better) models, but also put off other potential small companies from competing with those gigantic ones. Having said that, standardising the market with one particular model does have the advantage of concentrating the development of that particular model-based softwares, and hence improvements can be achieved much faster and as a result, cost of running/implementing/services will reduce significantly.

Having realised the needs for different application models, the following will be dedicated to those models and at the end of this section, a comparison of the four models will be made and see which one of them will best suit our future needs in general.

1.4.1 SQL databases

SQL (Structured Query Language) has been the standard data description and access language for relational databases for almost a decade, making it the core technology for client/server computing and dominating the client/server landscape today. It began as a declarative language for manipulating data using a few simple commands, however, as SQL applications moved to more demanding client/server environments, it became clear that just managing data wasn't enough. There was also a need to manage the functions that manipulated the data. And stored procedures, sometimes called "TP-lite", met the needs.

A stored procedure is a named collection of SQL statements and procedural logic that is complied, verified and stored in a server database. ***Sybase*** pioneered the concept of stored procedures and now virtually all SQL vendors support stored procedure along with other SQL extensions. The extensions are used to enforce data integrity, perform system maintenance and implement the server side of an application's logic.

The problem with the SQL standard is that there are just too many of them. There are at least eight efforts underway trying to create a standard based on SQL. *(ANSI alone is responsible for three standards, either published or in progress)* . And since SQL standards seem to lag vendor implementations by some years, almost everything that's interesting in client/server database technology is non-standard. This includes database administration, data replication, stored procedures, user-defined data types and the formats and protocols on networks. As a result, the lack of one widely accepted standard drives up the cost of databases and related tools and makes maintaining a client/server environment complex and difficult.

Although SQL suffers all this shortcomings, the fact that there are still so many people using it is because it is easy to create client/server applications in single-vendor/single-server environments. Many GUI tools make SQL applications easy to build, and most of all, it is familiar to the majority of programmers and users out there.

1.4.2 Transaction Processing (TP) monitors

In a simple client/server system, many clients issue requests and a server responses. This system may work for 50 or 100 clients, however, as the number of clients increases, the number of requests increases as well and eventually, it reaches the threshold limit of the system, i.e. the system crashes. Sadly, this is the case for most operating systems. Another technology, the TP monitors, is then developed to solve this problem.

Originally, TP monitor meant teleprocessing monitor - a program that multiplexed many terminals (clients) to a single central server. Over time, TP monitors took on more than just multiplexing and routing functions, and TP monitors came to mean transaction processing. TP monitors manage processes by breaking complex applications into pieces of code called *transactions*. A transaction can also be viewed as a set of actions that obeys the four so-called ***ACID*** properties - ***A***tomic, ***C***onsistent, ***I***solated, and ***D***urable. As for the middleware, TP monitors use some form of transactional RPC or peer-to-peer middleware.

TP monitors are probably overkilled in single-server/single-vendor departmental applications. That is probably one of the reasons why they have been so slow to take off. Moreover, vendors haven't come to grips yet with the realities of shrink wrapped software market, and they haven't been able to explain the advantages TP monitors offer. The modern client/server incarnations of TP monitors haven't dominant the

Ethernet era, but they'll definitely be playing a major role in the intergalactic era. It won't be long to assume that every machine on the network will have a TP monitors to represent it in global transactions. The intergalactic era will make those TP monitors advantages increasingly self-evident.

1.4.3 Groupware

Groupware comprises five foundation technologies geared to support collaborative work: multimedia management, work flow, E-mail, conferencing and scheduling. Groupware isn't just another downsized mainframe technology, it is indeed a new model of client/server computing. It helps users to collect unstructured data (e.g. text, images, faxes) into a set of documents.

Groupware has the advantage of document database management and makes effective use of E-mail, which is its preferred form of middleware . E-mail is one of the easiest ways for electronic processes to communicate with humans. Asynchronous by nature, it's a good match for the way business really work. E-mail is ubiquitous, with over 50 million globally interconnected electronic mailboxes.

Using work-flow to manages business processes is another revolutionary aspect of groupware. In a work-flow, data passes from one program to another in structured or unstructured client/server environments. Modern work-flow software electronically simulates real world collaborative activity. Work can be routed in ways that correspond to interoffice communications. A good work-flow packages lets you specify acceptance criteria for moving work from one stage to another. So work-flow brings the information to people who can act on it such that the work gets done by the right people.

Groupware provides many of the components we need for creating intergalactic client/server applications. The technology is also starting to encroach on its competitors' turf.

1.4.4 Distributed objects

Distributed-object technology promises the most flexible client/server systems. This is because it encapsulates data and business logic in objects that can roam anywhere on networks, run on different platforms, talk to legacy applications by way of object wrappers, and manage themselves and the resources they control. They are designed to become the currency of the intergalactic client/server era.

When it comes to standards, distributed-object technology is way ahead of all other client/server approaches. Since 1989, OMG has been busy specifying the architecture for an open software bus on which object components written by different vendors can interoperate across networks and operating systems.

The secret to OMG's success is that it defined how to specify an interface between a component and the object bus. Specifications are written in IDL , independent of any programming language. IDL becomes the contract that binds client to server components. The beauty of IDL is that it can easily be used to encapsulate existing applications. This way, existing applications need not be rewritten in order to take the full advantages of distributed-object technology.

In addition to defining the object bus, OMG has specified an extensive set of ORB-related services for creating and deleting objects, accessing them by names, defining the complex relationships among them. Later, OMG also defined a comprehensive set of services for transactional objects such that ordinary object can be created and maintained (e.g. make it transactional, lockable and persistent by having it inherit the appropriate services) using simple IDL entries.

Being reckon as the currency for the future intergalactic client/server era, distributed objects have to be powerful enough to replace all other client/server models in many aspects. For instance, TP monitors now have the advantages of better handling in transactions, concurrency, and scalability. So what can distributed objects do in order to over run TP monitors. The OMG has anticipated these problems long ago. The CORBA 2.0 aims to solve them by defining key object services, including transactions, concurrency, and relationships etc. Microsoft, with the help from Digital Equipment, has a rival solution known as COM (Commom Object Model) to solve the same problems.

1.5 Characteristics and Features in Client-Server Computing

Although there are various different configurations , different hardware and software platforms and even different network protocols in Client-Server Architecture. Generally they all possess certain characteristics and features that distinguish them from traditional mainframe computing environment.

1.5.1 Consists of a networked webs of small and powerful machines (both servers and clients)

Client-Server Computing uses local processing power-the power of desktop platform. It changes the way enterprise accesses, distributes, and uses data. With this approach, data is no longer under the tight control of Seniors Managers and MIS (Management of Information Systems) staff, it is readily available to middle-rank personel and staff. They can actively involved in the decision-making and operation on behalf of the company. The company becomes more flexible and gives a faster response to the changing business environment outside. In addition, if one machine goes down, the company will still function properly.

1.5.2 Open Systems

Another feature of Client-Server Computing is open systems, which means you can configure your systems, both software and hardware from various vendors as long as they stick to a common standard. In this way, company can tailor their system for their particular situation and needs, pick up and choose the most cost-effective hardware and software components to suit their tasks. For example, you can grab the data and run it through a spreadsheet from your desktop using the brand of computer and software tools that you're most comfortable with and get the job done in your own way.

1.5.3 Modularity

Since we are mixing softwares and hardwares of different natures together as a whole. All the software and hardware components are actually modular in nature. This modularity allows the system to expand and modernise to meet requirements and needs as the company grows. You can add or remove certain client or machine, implement some new application softwares and even add some hardware features without affecting the operation and functioning of the Client-Server System as a whole. Besides, as new computing platforms emerge, you can evalute new environments and system componenets in a modular fashion.

1.5.4 Cost Reduction and Better Utilisation of Resources

Potential cost savings prompt orgarnisations to consider Client-Server Computing. The combined base price of hardware(machines and networks) and software for Client/Server systems is often a tenth of the cost to that of mainframe computing. Furthermore,another feature of Client-Server Computing is able to link existing hardware and software applications and utilise them in an efficient way.

1.5.5 Complexity

Since the environment is typically heterogeneous and multivendor. The hardware platform and operating system of client and server are not usually the same. The biggest challenge to successfully implementing the system is to put together this complex system of hardware and software from multiple vendors. Therefore we need expertise not just for software, hardware or networks but expertise of all these fields and understand the interdependencies and interconnection. Sometimes when the system is down, it is extremely difficult to identify a bug or mistake, but there are just serveral culprits that might cause the problems. Futhermore we have to pay extra efforts and times in traning IS professionals to maintain this new environment in geographically dispersed location.

1.6 Main Applications

Client-Server Computing can also be categorized by their support function. The architecture of Client-Server Computing promotes group interaction, whether it is messages, mail, shared data, or shared applications. Users can be "closer" to one another. Users of an application can be anywhere on the network.

There are three main types of Client-Server Applications:

1.6.1 Database Access

This is the most important kind of application in Client-Server Computing. Applications in GUIs are written to access corporate data. These query- oriented applications provide a single window to the data of the organisation. In some cases these applications are read-only, in others they are read-write. Benefits of these systems are also ease of use and increased worker productivity. Productivity with these systems is measured by how easily workers can access the data they need to do the job. As we can see, we have a networked web of workstations for workers to access at any node, their productivity will increase tremendously. This kind of database system also provide transparent and consistent access to data wherever it is located.

1.6.2 Transaction-Processing Applications

Typical OLTP (Online-Transaction Processing Applications), also known as mission-critical applications, include order entry, inventory , and point-of-sale systems.

This kind of mission-critical system must run continuously. if it is unavailable even for a brief moment, the organizations will experience servere repercussions.

Examples are stock exchange system, air traffic control networks and airline reservation systems. The transactions are generated at the client and sent to the server for processing. The server may in turn, send one or more operations to other servers.

For a transaction to be considered complete, all operations must be successfully performed. If any operation of transaction cannot be completed, the operations that have taken effect must be reversed using a process called commit and rollback.

1.6.3 Office Systems

Many organizations are employing Client-Server Computing to improve interpersonal communications, both internally and externally. Many organizations are using their linked LANs as a network for enterprise-wide mail systems and workgroup applications. In this way, pesonnel in organization can improve coordination and actively participate in stategy formulation and decision-making of the company.

CHAPTER TWO

ADVANTAGES OF CLIENT/SERVER COMPUTING

2.1 General

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.

2.2.3 Sharing 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 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.

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 location or technology of the processors involved.

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 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 LAN market.

Step 1—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.

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.

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.

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 (SDE).

2.4 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 running 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.

2.5 Ways to Improve Performance

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

2.5.1 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.

2.5.2 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 100Mbs (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.

2.5.3 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.

2.6 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 .

2.6.1 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, FoxPro, 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 are 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.

2.6.2 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.

2.6.3 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.

2.7 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 DataEase—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 opportunity, programmers 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.

2.8 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

2.8.1 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.

2.8.2 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 impact 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/server 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. on.

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.

CHAPTER THREE

CLIENT/SERVER FUNDAMENTALS

3.1 Introduction

Simply stated, an object-oriented client/server Internet (OCSI) environment provides the IT infrastructure (i.e., middleware, networks, operating systems, hardware) that supports the OCSI applications–the breed of distributed applications of particular interest to us. The purpose of this chapter is to explore this enabling infrastructure before digging deeply into the details of application engineering and reengineering. Specifically, we review the following three core technologies of the modern IT infrastructure:

- Client/server that allows application components to behave as service consumers (clients) and service providers (servers).
- Internet for access to application components (e.g., databases, business logic) located around the world from Web browsers.
- Object-orientation to let applications behave as objects that can be easily created, viewed, used, modified, reused, and deleted over time.

In addition, we will attempt to answer the following questions:

- How are the key technologies combined to support the modern applications?
- What type of general observations can be made about the state of the art, state of the market, and state of the practice in OCSI environments?
- What are the sources of additional information on this topic?

Figure 3.1 will serve as a general framework for discussion. This framework, illustrates the role of the following main building blocks of OCSI environments:

- Client and server processes (applications) that represent the business logic as objects that may reside on different machines and can be invoked through Web services.
- Middleware that supports and enables the OCSI applications.
- Network services that transport the information between remote computers.
- Local services (e.g., database managers and transaction managers).
- Operating systems and computing hardware to provide the basic scheduling and hardware services

We will quickly scan these building blocks and illustrate their interrelationships in multivendor environments that are becoming common to support enterprisewide distributed applications.

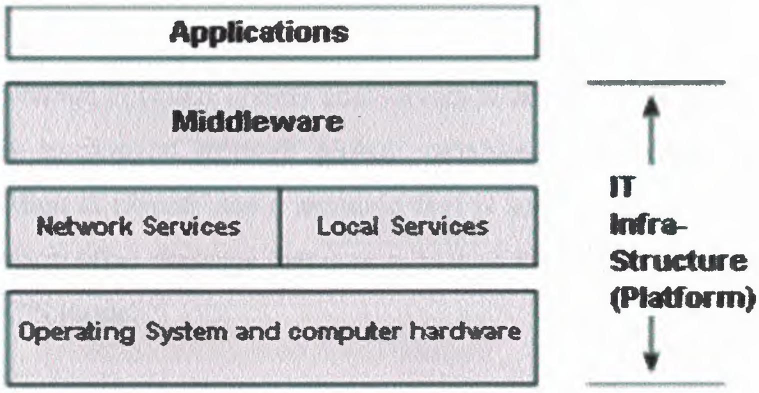

Figure 3.1. Object-Oriented Client/Server Internet Environments

3.2 Client/Server Fundamentals

3.2.1 Definitions

Client/server model is a concept for describing communications between computing processes that are classified as service consumers (clients) and service providers (servers). Figure 3.2 presents a simple C/S model. The basic features of a C/S model are:

1. Clients and servers are functional modules with well defined interfaces (i.e., they hide internal information). The functions performed by a client and a server can be implemented by a set of software modules, hardware components, or a combination thereof. Clients and/or servers may run on dedicated machines, if needed. It is unfortunate that some machines are called "servers." This causes confusion (try explaining to an already bewildered user that a client's software is running on a machine called "the server"). We will avoid this usage as much as possible.

2. Each client/server relationship is established between two functional modules when one module (client) initiates a service request and the other (server) chooses to respond to the service request. Examples of service requests (SRs) are "retrieve customer name," "produce net income in last year," etc. For a given service request, clients and servers do not reverse roles (i.e., a client stays a client and a server stays a server). However, a server for SR R1 may become a client for SR R2 when it issues requests to another server (see Figure 3.2). For example, a client may issue an SR that may generate other SRs.

3. Information exchange between clients and servers is strictly through messages (i.e., no information is exchanged through global variables). The service request and additional information is placed into a message that is sent to the server. The server's response is similarly another message that is sent back to the client. This is an extremely crucial feature of C/S model.

The following additional features, although not required, are typical of a client/server model:

4. Messages exchanged are typically interactive. In other words, C/S model does not support an off-line process. There are a few exceptions. For example, message queuing systems allow clients to store messages on a queue to be picked up asynchronously by the servers at a later stage.

5. Clients and servers typically reside on separate machines connected through a network. Conceptually, clients and servers may run on the same machine or on separate machines. In this book, however, our primary interest is in *distributed client/server systems* where clients and servers reside on separate machines.

The implication of the last two features is that C/S service requests are real-time messages that are exchanged through network services. This feature increases the appeal of the C/S model (i.e., flexibility, scalability) but introduces several technical issues such as portability, interoperability, security, and performance.

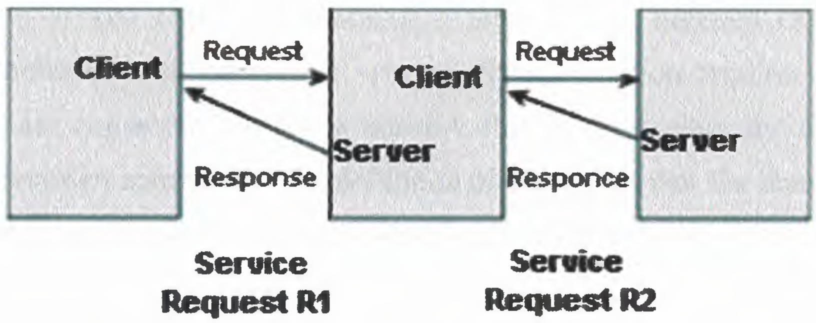

Figure 3.2. Conceptual Client/Server Model

3.2.2 Client/Server–A Special Case of Distributed Computing

Figure 3.3 shows the interrelationships between distributed computing and client/server models. Conceptually, client/server model is a special case of distributed-computing model.

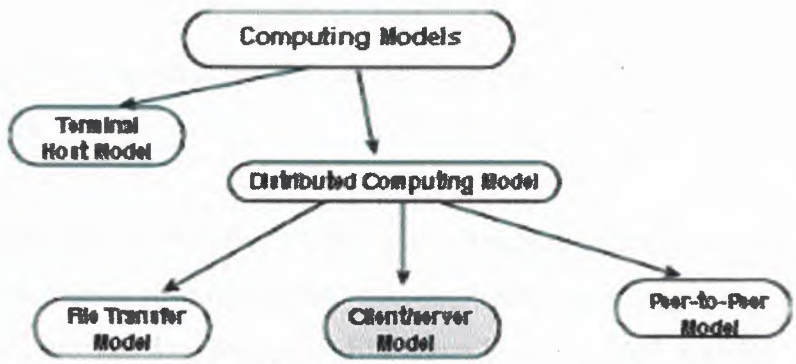

Figure 3.3. Interrelationships between Computing Models

A Distributed Computing System (DCS) is a collection of autonomous computers interconnected through a communication network to achieve business functions. Technically, the computers do not share main memory so that the information cannot be transferred through global variables. The information (knowledge) between the computers is exchanged only through messages over a network.

The restriction of no shared memory and information exchange through messages is of key importance because it distinguishes between DCS and shared memory multiprocessor computing systems. This definition requires that the DCS computers are connected through a network that is responsible for the information exchange between computers. The definition also requires that the computers have to work together and cooperate with each other to satisfy enterprise needs.

3.2.3 Client/Server Architectures

Client/server architecture provides the fundamental framework that allows many technologies to plug in for the applications of 1990s and beyond. Clients and servers typically communicate with each other by using one of the following paradigms:

- Remote Procedure Call (RPC). In this paradigm, the client process invokes a remotely located procedure (a server process), the remote procedure executes and sends the response back to the client. The remote procedure can be simple (e.g., retrieve time of day) or complex (e.g., retrieve all customers from Chicago who have a good credit rating). Each request/response of an RPC is treated as a separate unit of work, thus each request must carry enough information needed by the server process. RPCs are supported widely at present.

- Remote Data Access (RDA). This paradigm allows client programs and/or end-user tools to issue ad hoc queries, usually SQL, against remotely located databases. The key technical difference between RDA and RPC is that in an RDA the size of the result is not known because the result of an SQL query could be one row or thousands of rows. RDA is heavily supported by database vendors.

- Queued Message Processing (QMP). In this paradigm, the client message is stored in a queue and the server works on it when free. The server stores ("puts") the response in another queue and the client actively retrieves ("gets") the responses from this queue. This model, used in many transaction processing systems, allows the clients to asynchronously send requests to the server. Once a request is queued, the request is processed even if the sender is disconnected (intentionally or due to a failure). QMP support is becoming commonly available.

Initial implementations of client/server architecture were based on the "two-tiered" architectures shown in Figure 3.4 (a) through Figure 3.4 (e) (these architectural configurations are known as the "Gartner Group" configurations). The first two architectures (Figure 3.4 (a) and Figure 3.4 (b) are used in many presentation intensive applications (e.g., XWindow, multimedia presentations) and to provide a "face lift" to legacy applications by building a GUI interface that invokes the older text-based user interfaces of legacy applications. Figure 3.4 (c) represents the distributed application program architecture in which the application programs are split between the client and server machines, and they communicate with each other through the remote procedure call (RPC) or queued messaging middleware. Figure 3.4 (d) represents the remote data architecture in which the remote data is typically stored in a "SQL server" and is accessed through ad hoc SQL statements sent over the network. Figure 3.4 (e) represents the case where the data exist at client as well as server machines (distributed data architecture).

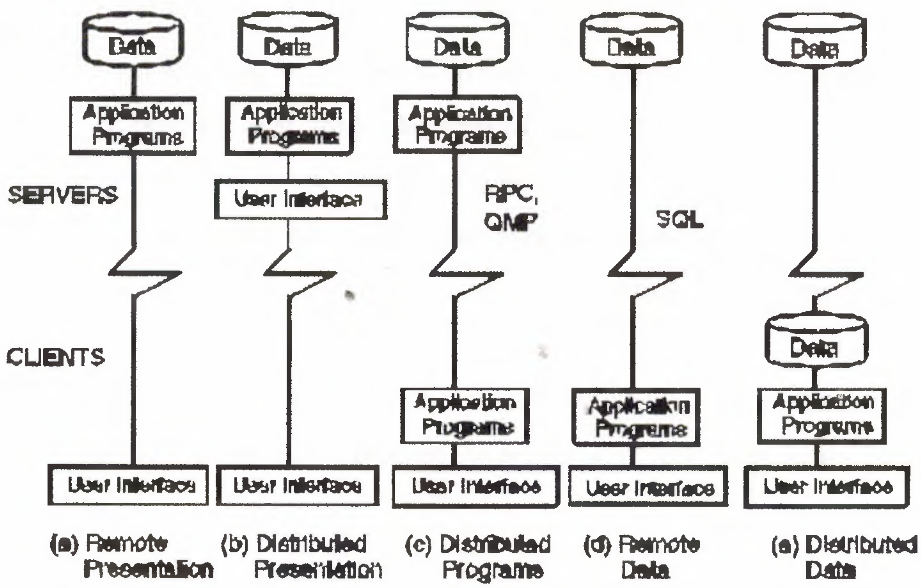

Figure 3.4. Traditional Client/Server Architectures

Although a given C/S application can be architected in any of these configurations, the remote data and distributed program configurations are used heavily at present. The remote data configuration at present is very popular for departmental applications and is heavily supported by numerous database vendors (as a matter of fact this configuration is used to represent typical two-tiered architectures that rely on remote SQL). Most data warehouses also use a remote data configuration because the data warehouse tools can reside on user workstations and issue remote SQL calls to the data warehouse (we will discuss data warehouses in Chapter 10). However, the distributed programs configuration is very useful for enterprisewide applications, because the application programs on both sides can exchange information through messages.

3.3 Internet and the World Wide Web (WWW)

Origin of the Internet is the ARPANET (Advanced Research Projects Agency Network) that was initiated in 1969 to support researchers on DOD (Department of Defense) projects. For many years, Internet had been used mainly by scientists and programmers to transfer files and send/receive electronic mail. The users of Internet relied on text-based user interfaces and tedious commands to access remote computing resources. In 1989, this changed with the introduction of World Wide Web (WWW), commonly referred to as the Web. The Web has been a major contributor in turning the Internet, once an obscure tool, into a household word. Why? Mainly because the Web allows users to access, navigate, and share information around the globe through GUI clients ("Web browsers") that are available on almost all computing platforms. The Web browsers allow users to access information that is linked through hypermedia links.

Thus a user transparently browses around, or "surfs" around, different pieces of information that are located on different computers in different cities and even in different countries.

3.3.1 Internet, Intranets, and Extranets

Simply stated, Internet is a network of networks. Technically, however, Internet is a collection of networks based on the IP (Internet Protocol) stack. This protocol stack, initially referred to as the DOD (Department of Defense) or ARPANET Protocol Suite and commonly referred to as the TCP/IP (Transmission Control Protocol/Internet Protocol) was designed to support e-mail, file transfer, and terminal emulation for ARPANET users. The services and protocols supported by IP have dramatically grown in popularity and have become the de facto standards for heterogeneous enterprise networks. At present, the term Internet is used to symbolize the IP (or loosely speaking, TCP/IP) networks in the following situations:

- Public Internet, or just the Internet, that is not owned by any single entity–it consists of many independent IP networks that are tied together loosely. Initially, the public Internet was used to tie different university networks together. With time, several commercial and private networks have joined the public Internet. The computers on the public Internet have publicly known Internet Protocol (IP) addresses that are used to exchange information over the public Internet (see discussion on addressing below). The public Internet at present consists of thousands of networks.

- Private Internets, or Intranets, are the IP networks that are used by corporations for their own businesses. Technically, an Intranet is the same as the public Internet, only smaller and privately owned (thus hopefully better controlled and more secure). Thus any applications and services that are available on the public Internet are also available on the Intranets. This is an important point for WWW, because many companies are using WWW technologies on their Intranets for internal applications (e.g., employee information systems).

- Business-to-business Internets, or Extranets, are the TCP/IP networks that are used for business-to-business activities. Technically, an Extranet is the same as the public Internet but is better controlled and more secure. Many electronic commerce applications between business partners are beginning to use Extranets. Any applications and services that are available on the public Internet are also available on the Extranets.

Basically: Internet = Public Internet + Intranets + Extranets

The following protocols (the first three belong to the original DOD Suite) are among the best known application protocols in the Internet (see Figure 2.6):

- Telnet: This protocol is used to provide terminal access to hosts and runs on top of TCP.
- File Transfer Protocol (FTP): This TCP-based protocol provides a way to transfer files between hosts on the Internet.
- Simple Mail Transfer Protocol (SMTP): This TCP-based protocol is the Internet electronic mail exchange mechanism.
- Trivial File Transfer Protocol (TFTP): This UDP-based protocol also transfers files between hosts, but with less functionality (e.g., no authorization mechanism). This protocol is typically used for "booting" over the network.
- Network File System (NFS) Protocol: This UDP-based protocol has become a de facto standard for use in building distributed file systems through transparent access.
- Xwindow: This is a windowing system that provides uniform user views of several executing programs and processes on bit-mapped displays. Although Xwindow is supposedly network independent, it has been implemented widely on top of TCP.
- SUN Remote Procedure Call (RPC): This protocol allows programs to execute subroutines that are actually at remote sites. RPCs, similar to Xwindow, are supposedly network independent but have been implemented widely on top of TCP. SUN RPC is one of the oldest RPCs. Examples of other RPCs are OSF, DCE, RPC, and Netwise RPC.
- Domain Naming Services: This protocol defines hierarchical naming structures that are much easier to remember than the IP addresses. The naming structures define the organization type, organization name, etc.
- SNMP (Simple Network Management Protocol): This is a protocol defined for managing (monitoring and controlling) networks.
- Kerberos: This is a security authentication protocol developed at MIT.
- Time and Daytime Protocol: This provides machine-readable time and day information.

The World Wide Web (WWW) has introduced additional application protocols and services. For example, the Web browsers, the Web servers, and the HTTP protocol used in WWW reside on top of the IP stack (see next section). As the use of Internet grows, more services and protocols for the IP application layer will emerge.

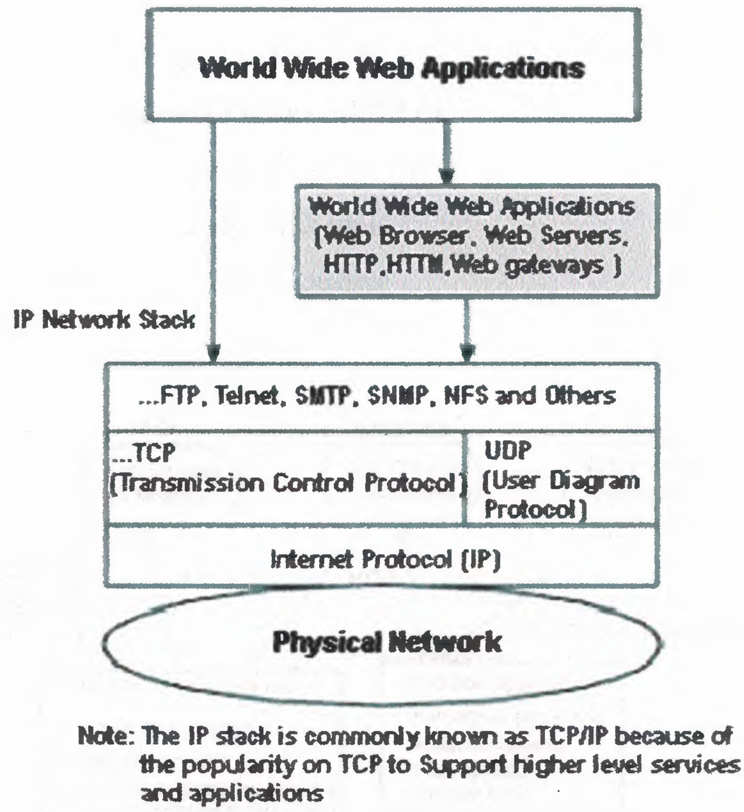

Figure 3.5. Technical View of Internet and World Wide Web

3.3.2 Overview of World Wide Web

World Wide Web (WWW) is a wide area information retrieval project that was started in 1989 by Tim Berners-Lee at the Geneva European Laboratory for Particle Physics (known as CERN, based on the laboratory's French name) [Berners-Lee 1993] and [Berners-Lee 1996]. The initial proposal suggested development of a "hypertext system" to enable efficient and easy information sharing among geographically separated teams of researchers in the High Energy Physics community.

Technically speaking, WWW is a collection of middleware that operates on top of IP networks (i.e., the Internet). Figure 3.5 shows this layered view. The purpose of the WWW middleware is to support the growing number of users and applications ranging from entertainment to corporate information systems. Like many other (successful) Internet technologies, the WWW middleware is based on a few simple concepts and technologies such as the following (see Figure 3.6):

- Web servers
- Web browsers
- Uniform Resource Locator (URL)
- Hypertext Transfer Protocol (HTTP)
- Hypertext Markup Language (HTML)
- Web navigation and search tools
- Gateways to non-Web resources

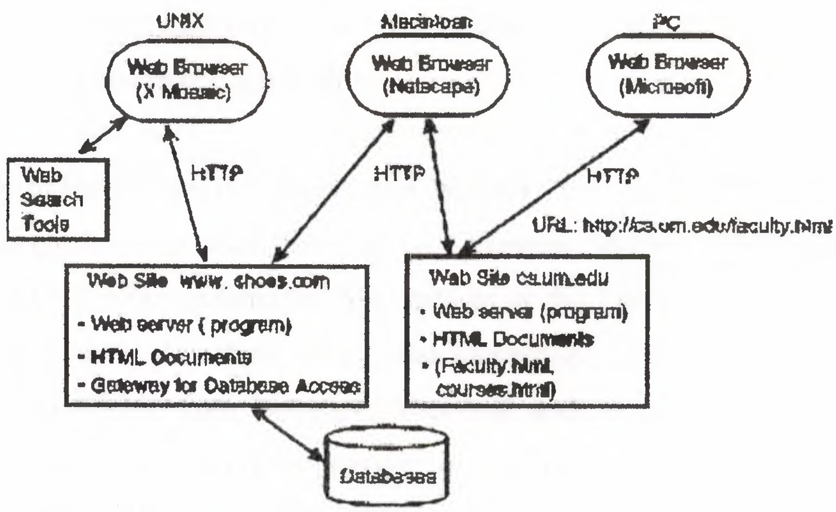

Figure 3.6. Conceptual View of World Wide Web

Let us briefly review these components and show how they tie with each other through an example.

Web sites provide the content that is accessed by Web users. Web sites are populated and in many cases managed by the content providers. For example, Web sites provide the commercial presence for each of the content providers doing business over the Internet. Conceptually, a Web site is a catalog of information for each content provider over the Web. In reality, a Web site consists of three types of components: a Web server (a program), content files ("Web pages"), and/or gateways (programs that access non-Web content). A Web server is a program (technically a server process) that receives calls from Web clients and retrieves Web pages and/or receives information from gateways (we will discuss gateways later). Once again, a Web user views a Web site as a collection of files on a computer, usually a UNIX or Windows NT machine. In many cases, a machine is dedicated/designated as a Web site on which Web accessible contents are stored. As a matter of convention, the entry point to a Web site is a "home page" that advertises the company business. Very much like storefront signs in a shopping mall, the home pages include company logos, fancy artwork for attention, special deals, overviews, pointers to additional information, etc. The large number of Web sites containing a wide range of information that can be navigated and searched transparently by Web users is the main strength of WWW. Figure 3.6 shows two Web sites–one for a shoe shop (www.shoes.com) and the other for a computer science department for a university (cs.um.edu).

Web browsers are the clients that typically use graphical user interfaces to wander through the Web sites. The first GUI browser, Mosaic, was developed at the National Center for Supercomputer Applications at the University of Illinois. Mosaic runs on PC Windows, Macintosh, UNIX, and Xterminals. At present, Web browsers are commercially available from Netscape, Microsoft and many other software/freeware providers. These Web browsers provide an intuitive view of information where hyperlinks (links to other text information) appear as underlined items or highlighted text/images. If a user points and clicks on the highlighted text/images, then the Web browser uses HTTP to fetch the requested document from an appropriate Web site. Web browsers are designed to display information prepared in a markup language, known as HTML. Even though these are different browsers residing on different machines, they all use the same protocol (HTTP) to communicate with the Web servers (HTTP compliance is a basic requirement for Web browsers).

CHAPTER FOUR

COMPONENTS OF CLIENT/SERVER COMPUTING AND ITS APPLICATIONS-THE CLIENT

4.1 General

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 client workstation. There is no specific technological characteristic of a client.

During the past 15 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 15 years. In 1984, the performance and reliability of remote 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 performance 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.

4.2 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). 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.

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 analysis can reside on the client workstation.

A client workstation uses a local operating system to host both basic services and 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—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.

4.3 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 911 emergency telephone number and dispatches fire, police, ambulance, or emergency vehicles to an incident.

4.4 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?

1. 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.

2. 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 all NOS and UNIX network file system (NFS) vendors. The simplicity of executing standard calls to a virtual network of services is its main advantage.

4.4.1 Remote Procedure Call (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.

4.4.2 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.

4.4.3 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.

4.4.4 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 remaining services and applications functions to the client workstation or X-terminal.

4.4.5 Utility Services

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

4.4.6 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.

4.4.7 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.

4.4.8 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.

4.4.9 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.

4.4.10 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.

4.4.11 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.

4.4.12 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 running 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.

4.4.13 Common Object Request Broker Architecture (CORBA)

CORBA is 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.

4.5 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!

CHAPTER FIVE

COMPONENTS OF CLIENT/SERVER COMPUTING AND ITS APPLICATIONS-THE SERVER

5.1 General

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.

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.

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 internetworking.

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.

Microsoft's LAN Manager NOS and its several derivatives—including IBM Lan Server, HP LAN Manager/UX 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.

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.

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.

5.2 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 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.

5.3 Server Functionality in Detail

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

5.3.1 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.

5.3.2 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. 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 colors, 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.

5.3.3 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.

5.3.4 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. Figure 5.1 charts the evolution of database technology from the first computers in the late 1950s to the object-oriented database technologies that are becoming prevalent in the mid-1990s.

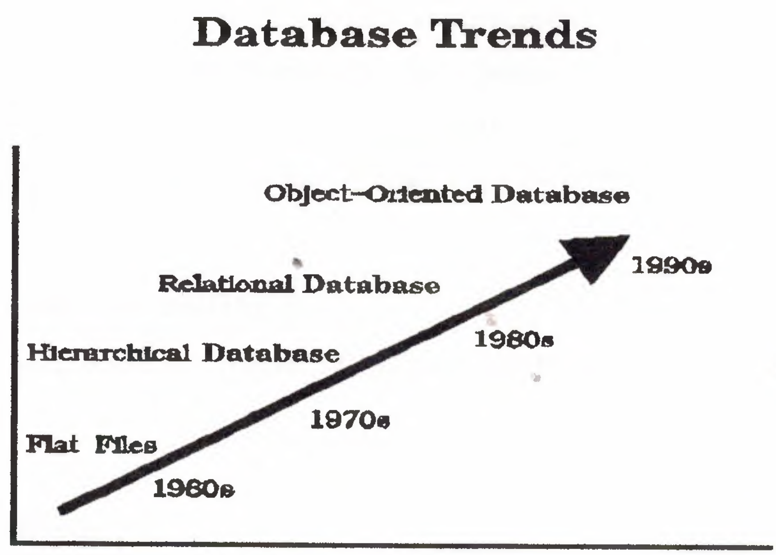

Figure 5.1. Database Trends

5.3.5 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. This chapter provides a complete discussion of connectivity issues in the client/server model.

5.3.6 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.

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.

5.4 The Network Operating System

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

5.4.1 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 running 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 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. Figure 4.2 shows the major components of the NetWare architecture, illustrating client and server functions.

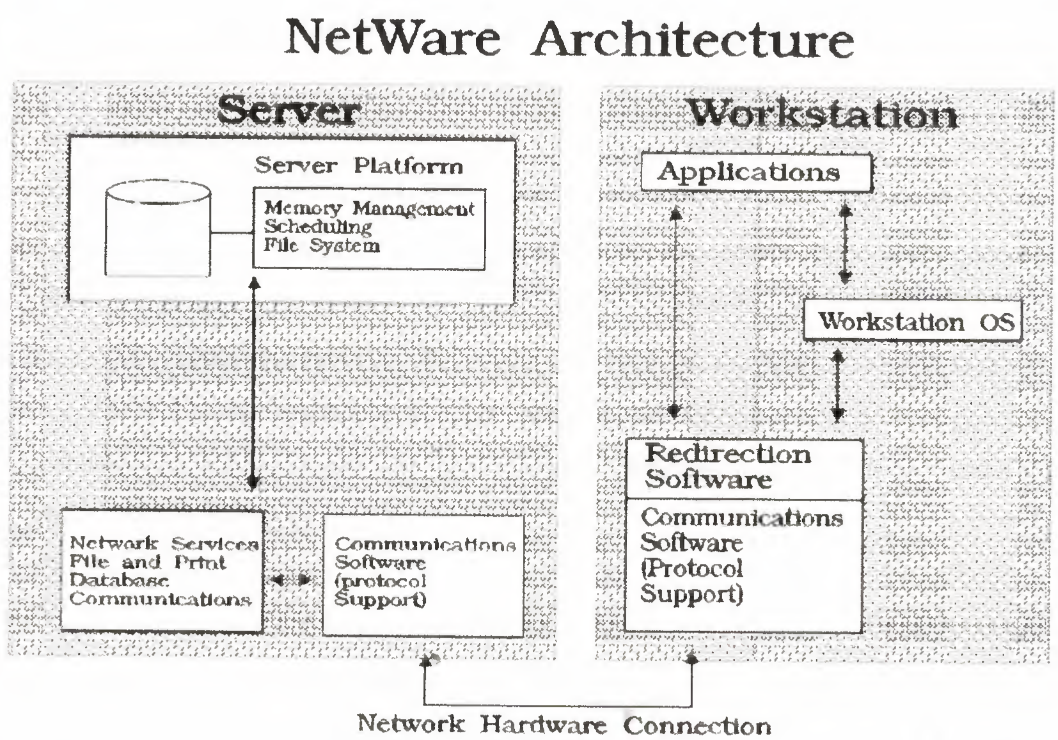

Figure 5.2. Netware Architicture

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 (ODI) and NetWare Streams—enable other vendors to develop products for the NetWare environment. This facilitates its integration into other platforms

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.

5.4.2 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. Then 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 did not find Advanced Server an answer to their dreams before 1995.

5.4.3 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.

5.4.4 Banyan VINES

Banyan VINES provides basic file and print services similar to those of Novell and Lan Manager.

VINES incorporates a facility called StreetTalk 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.

5.4.5 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 (NDIS) drivers.

5.5 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.

a. 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. The products will enable UNIX to simultaneously access information from both a NetWare and a UNIX server.

b. 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 for the 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.

c. 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 took 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 installed by many organizations as their server of choice.

d. MVS

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—SNA—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. MVS 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.

e. 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.

f. 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, AIX. Finally, in 1989, an agreement was reached on the basic UNIX kernel, shell functions, and APIs.

5.6 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 is 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/MVS data storage facility.

CHAPTER SIX

COMPONENTS OF CLIENT/SERVER COMPUTING AND ITS APPLICATIONS-CONNECTIVITY

6.1 General

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 "sneakernet" (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.

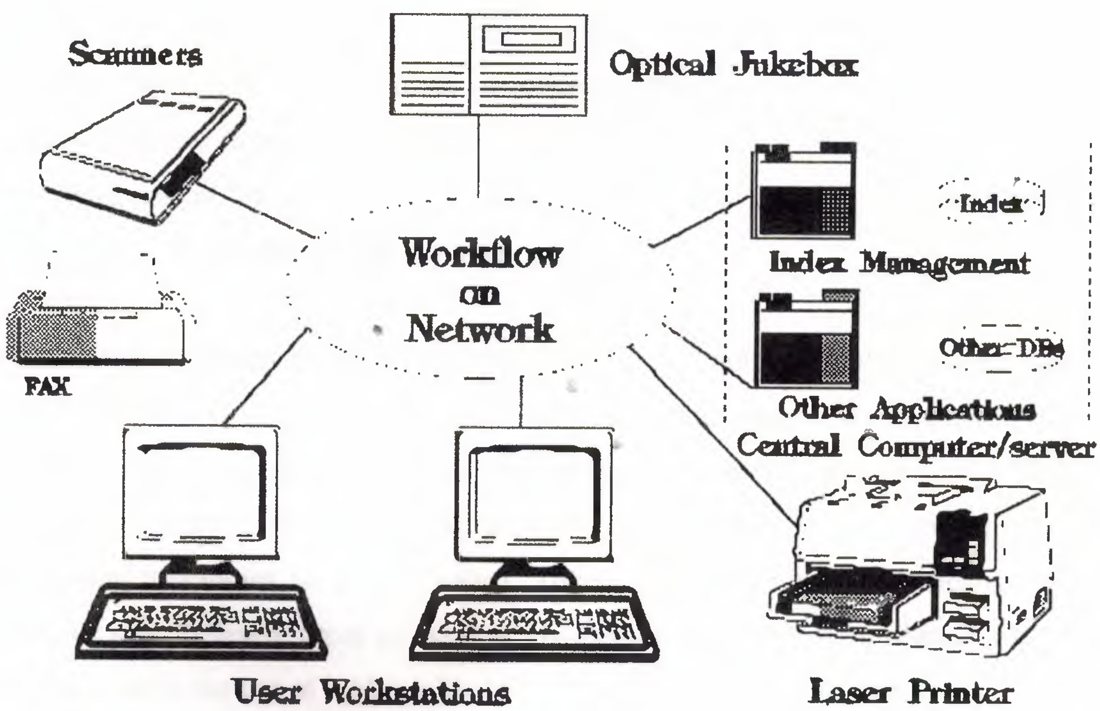

Figure 6.1. Workflow on Network

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 turn 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.

Figure 6.2 shows an example for using a client/server architecture and a mainframe DB2 database for data archiving . This example provides a high-level view of the system components.

The Slope Inclinometer System (SLOPE) was developed to assist the geotechnical engineers in determining ground movement near the edge of the pit in order to ensure the safety and productivity of the huge draglines used in the extraction of ore from the mine. This is accomplished through the capture, analysis, and reporting of ground movement data in a responsive LAN environment, using intelligent workstations equipped with a graphical user interface (GUI).

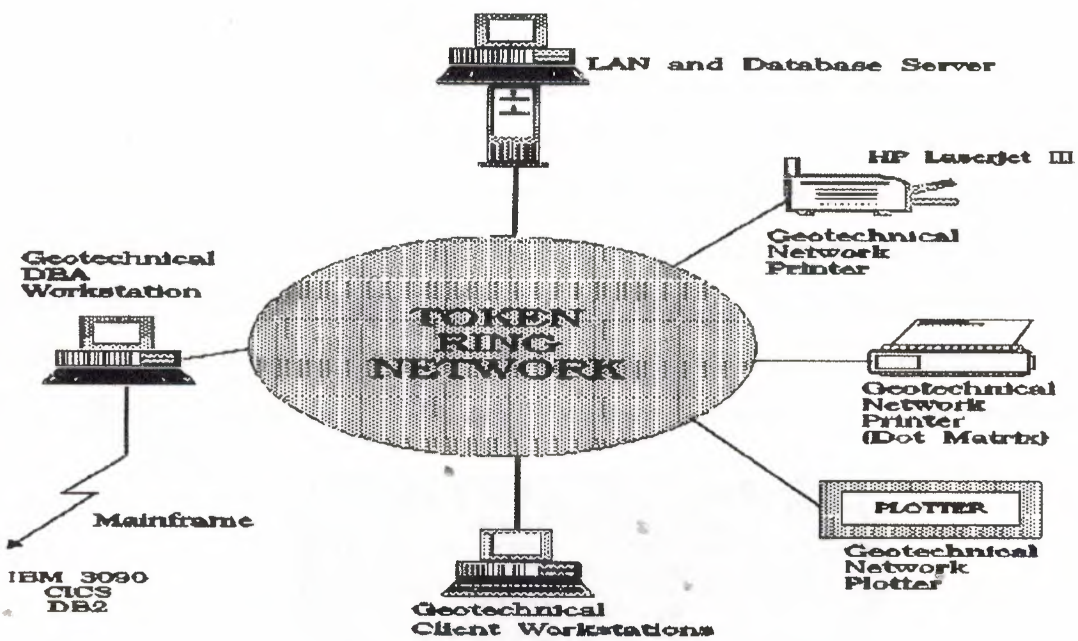

Figure 6.2. The components of Syncrude Canada's Slope Inclinometer System

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.

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.

6.2 Open Systems Interconnect

The OSI reference model shown in Figure 6.3 provides an industry standard framework for network and system interoperability. 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. The OSI model provides the framework definition for developers attempting to create interoperable products. Because many products are not yet OSI-compliant, there often is no direct correspondence between the OSI model and reality.

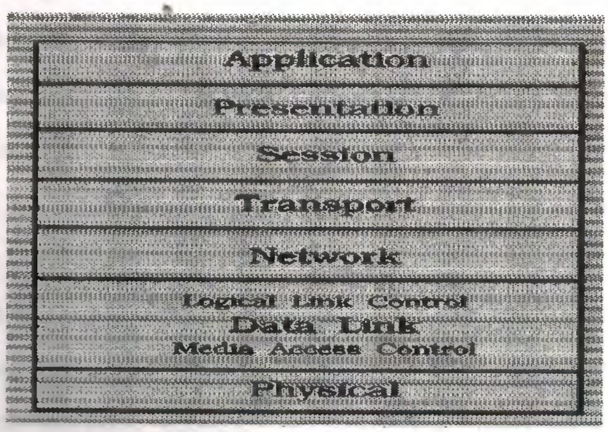

Figure 6.3. the seven-layer OSI model.

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

a. 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.

b. Data Link 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.

c. 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.

d. 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 destination) can share a single network address.

e. 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.

f. 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.

g. 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.

6.3 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.

6.3.1 LAN Cabling

One of the most important and most overlooked parts of LAN implementation today is the physical cabling plant. 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 Ring, FDDI, and future ATM LANs.

Cabling standards include RG-58 A/U coaxial cable (thin-wire 10Base2 Ethernet), IBM Type 1 (shielded, twisted pair for Token Ring), unshielded twisted pair (UTP for 10BaseT Ethernet or Token Ring) and Fiber Distributed Data Interface (FDDI for 10BaseT 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.

6.3.2 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.

10BaseT 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 10BaseT the most popular standards-based network topology.

6.3.3 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 100Mbps standard for both Token Ring and Ethernet networks. This technology, called 100VG-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. IBM will be submitting a proposal to make the 100VG-AnyLAN technology a part of IEEE's 802.12 (or 100Base-VG) standard, which currently includes only Ethernet. 100VG-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 SynOptics, 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.

6.3.4 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 of FDDI is in a star configuration, and provides support for distances of up 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 of the 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.

6.3.5 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.

6.3.6 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.

Figure 6.4 presents the results of a recent study of installation plans for Ethernet, Token Ring, and FDDI. The analysis predicts a steady increase in planned Token Ring installations from 1988 until the installed base is equivalent in 1998. 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 1998 ATM will dominate all new installations and will gradually replace existing installations by 1999.

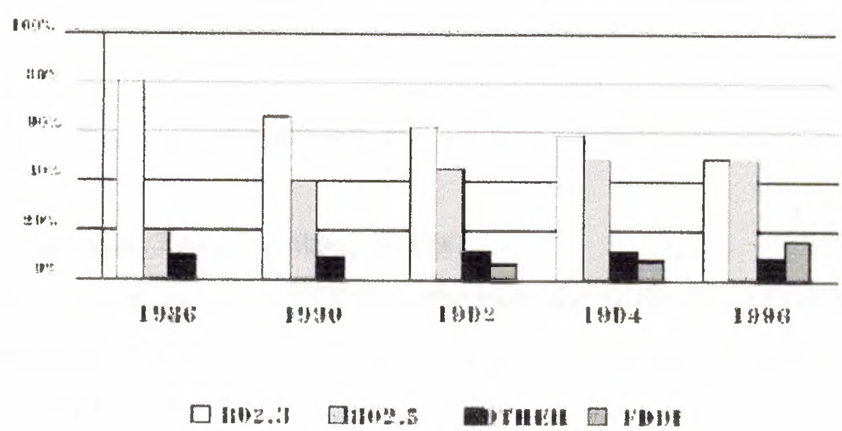

Figure 6.4. LAN-Host connection

Figure 6.5 illustrates the interoperability possible today with routers from companies such as Cisco, Proteon, Wellfleet, Timeplex, Network Systems, and 3-Com. Most large organizations should provide support for the three different protocols and install LAN topologies similar to the one shown in Figure 6.4. Multiprotocol routers enable LAN topologies to be interconnected.

LAN topologies similar to the one shown in Figure 6.4. Multiprotocol routers enable LAN topologies to be interconnected.

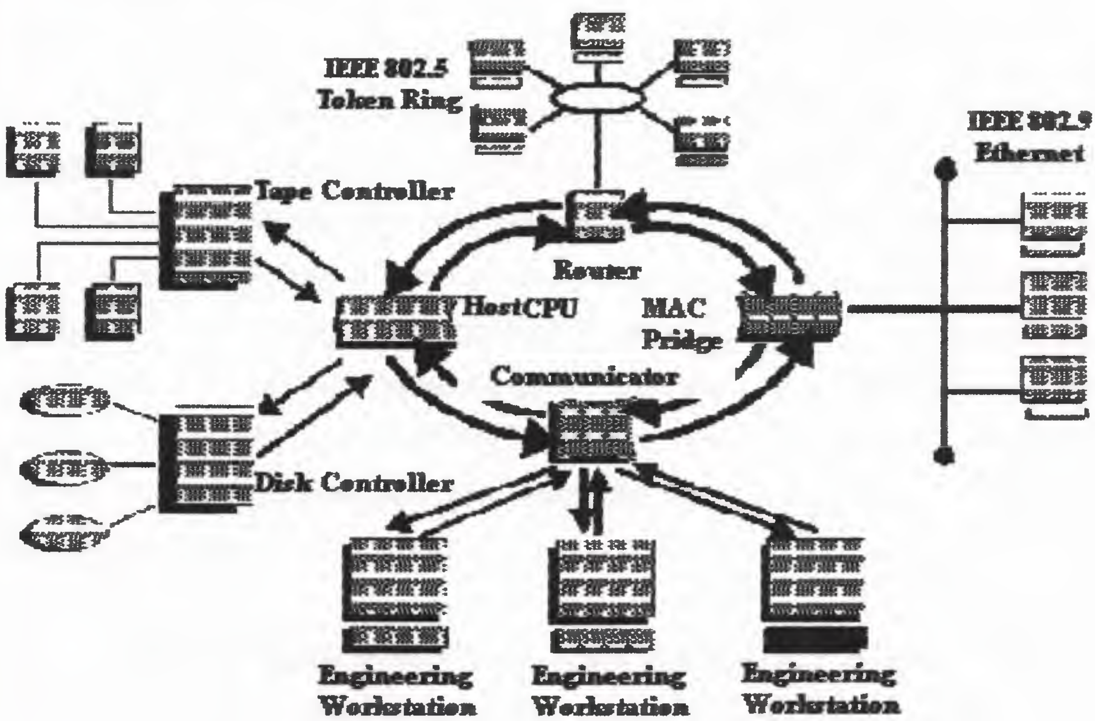

Figure 6.5. FDDI interoperability

6.3.7 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.

network managers with the required flexibility to respond promptly to business change and new applications.

6.3.8 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 10BaseT 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. (See Figure 6.6.)

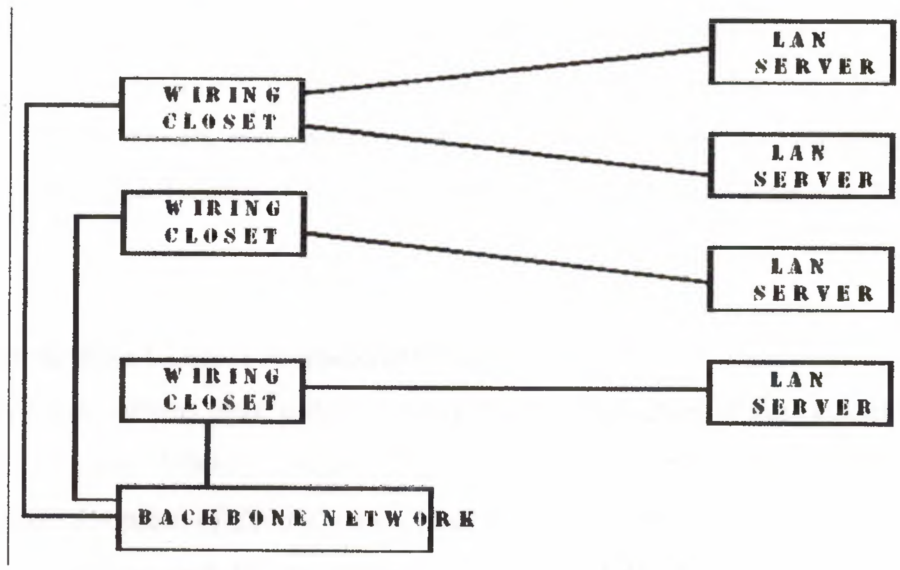

Figure 6.6. Distribution of LAN servers

6.3.9 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.

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, FDDI, and from 8 to 10 different protocols.

6.3.10 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/Internet 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.

6.3.10.1 TCP/IP'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.

6.3.10.2 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.

6.3.10.3 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.

6.3.10.4 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.

6.3.10.5 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.

6.3.11 Simple Network Management Protocol (SNMP)

SNMP provides intelligence and services to effectively manage an internetwork. It has 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).

6.3.12 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.

6.3.13 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.

6.4 Interprocess Communication

At the top of the OSI model, interprocess communications (IPCs) define the format for application-level interprocess communications. In the client/server model, there is always a need for interprocess communications. IPCs 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.

6.4.1 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.

6.4.2 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 of X'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.

6.4.3 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.

6.4.4 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.

6.4.5 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.

6.4.6 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. 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.

6.4.7 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.

6.4.8 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.

6.4.9 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.

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.

We 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. And we 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 tools into their client/server applications should insist that all desktop products they acquire be DDE-capable.

6.4.10 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.

CONCLUSION

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 WANs (including the Internet).

Client/server computing is a common networking model which enables many users to access information in an efficient manner. Generally, the user's computer is called the client and the machine that contains the information being accessed is called the server.

The client computer runs an application called a client program. A client program enables a user to send a request for information to the server and read the results that the server sends back. The server computer runs a server program which processes requests and sends results back to the client. Most Internet transactions, such as FTP, e-mail and accessing web pages are based on client/server networking.

Client/server is just one approach to distributed computing. The client/server model has been popular for a long time, but recently Peer-to-Peer networking has re-emerged as a viable alternative. This is because of the peer-to-peer (P2P) networks eliminate the need for servers and allows all computers to communicate and share resources as peers. Many popular client applications like ICQ and Napster rely on P2P technology.

Client-server growth will continue throughout this decade and beyond. Although C-S technologies and methodologies are still evolving, CIS curricula need to begin the client-server learning processing by gaining some initial hands-on experiences.

No longer a fad, client/server computing is the information technology mainstream for the 1990s. It may be too early to predict the demise of traditional mainframe computing; generally, new technologies supplement rather than displace old ones. But even though large institutional databases will continue to reside on institutional mainframes and users will continue to access them using keyboards and monitors, the systems connecting user and data will undergo dramatic structural changes over the next few years.

REFERENCES

[1] Patrick N. Smith , Client/Server Computing, SHL Systemhouse, Inc., United States of America, January 1994.

[2] Buzzard, James. The Client-Server Paradigm: Making Sense Out of the Claims, Data Based Advisor, August 1990.

[3] McGoveran, David. Evaluating an RDBMS for Client/Server Applications. Alternative Technologies, 1992.

[4] Adler, R. M. "Distributed Coordination Models for Client/Sever Computing." Computer 28, April 1995, pp. 14-22.

[5] Architecture. A Comparison of Two-Tier and Three-Tier Systems." *Information Systems Management Magazine"*, Spring 1996.

[6] Brandel, M. Global Marine. *"Client/Server Journal"*, 1995, pp. 25-27.

[7] Garner, R. Hughes Space communications. *"Client/Server Journal"*, 1995, pp. 28-29.

[8] Schatz, W. Fannie Mae. *"Client/Server Journal"*, 1995, pp.22-23.

[9] "Remapping the Computing Environment for the 90s: The Impact of Client/Server," Sybase Executive Summary, Winter 1993.

[10] Client/Server Computing with the AS/400. A Duke Communication International White Paper published as a supplement to News 3X/400, 1992.

[11] Lloyd Taylor, "comp. Client-ServerFAQ maintainer", "http://www.wp.com/Lloyd.Taylor". 1999.

[12] Schussel, George. *"Client/Server Past, Present, and Future"*, *"http://www.dciexpo.com/geos"*, 1995.