# CST426 - CLIENT SERVER ARCHITECTURE

# Module – 3
# Client/Server Network

- Client
  - Services
  - Request for services
  - RPC
  - Windows services
  - Print services
  - Remote boot services
  - other remote services
  - Utility Services
  - Dynamic Data Exchange (DDE)
  - Object Linking and Embedding (OLE)
  - Common Object Request Broker Architecture (CORBA).

- Server
  - Detailed server functionality
  - Network operating system
  - Available platforms
  - Server operating system.

# Client Services

- Some of the **main services** that client performs :
- Responsible for managing the user interface.
- Provides presentation services.
- Accepts and checks the syntax of user inputs.
- Acts as a consumer of services provided by one or more server processors.
- Processes application logic.
- The role of the client process can be further extended at the client by adding logic that is not implemented in the host server application.
- Local editing, automatic data entry, help capabilities, and other logic processes can be added in front of the existing host server application.
- Generates database request and transmits to server.
- Passes response back to server.

- Some of the other important services that are directly or indirectly attached with the client services are given below:

  (*a*) *Inter process communication.*

  (*b*) *Remote services.*

  (*c*) *Window services.*

  (*d*) *Dynamic data exchange.*

  (*e*) *Object linking and embedding(OLE).*

  (*f*) *Common object request broker architecture (CORBA).*

  (*g*) *Print/Fax services.*

  (*h*) *Database services.*

# Request for Service

- Client workstations request services from the attached server.
- Whether this server is the same processor or a network processor, the application format of the request is the same.
- **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.***
- 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.
    2. The NOS requester software constructs the remote procedure call (RPC) to include the API call to the NOS server.
    3. The NOS server then processes the request as if it were executed locally and ships the response back to the application.

# Remote Procedure Call (RPC)

- RPC is a powerful technique for constructing distributed, client-server based applications.
- It allows programs on different machines to interact using simple procedure call or return semantics, just as if the two programs were on the same machine.
- It is based on extending the notion of local procedure calling, so that the called procedure need not exist in the same address space as the calling procedure.
- The two processes may be on the same system, or they may be on different systems with a network connecting them. ie, the procedure call is used to access remote services.

- *How RPC Works:*
- An RPC mechanism is analogous to a function call.
- Like a function call, when an RPC is made, the calling arguments are passed to the remote procedure and the caller waits for a response to be returned from the remote procedure.
- Fig illustrates the general architecture of remote procedure call mechanism that takes place during an RPC call between two networked systems.
- The client makes a procedure call that sends a request to the server and waits.
- The thread is blocked from processing until either a reply is received, or it times out.
- When the request arrives, the server calls a dispatch routine that performs the requested service, and sends the reply to the client.
- After the RPC call is completed, the client program continues. RPC specifically supports network applications.

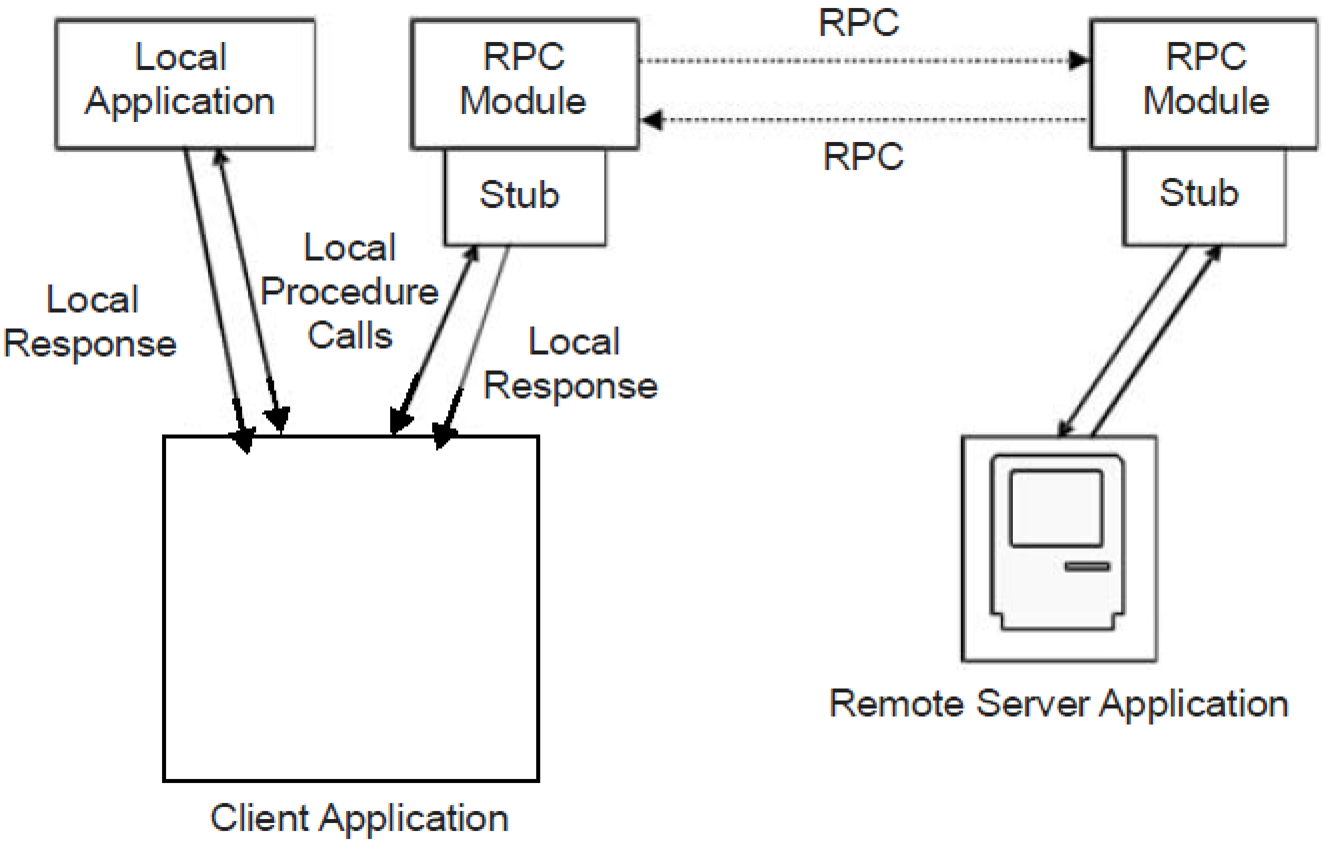


Fig. 5.7: RPC Mechanism

- A remote procedure is uniquely identified by the triple: **program number, version number, procedure number**
- The program number identifies a group of related remote procedures, each of which has a unique procedure number.
- A program may consist of one or more versions.
- Each version consists of a collection of procedures which are available to be called remotely.
- Each version contains a number of procedures that can be called remotely.
- Each procedure has a procedure number.

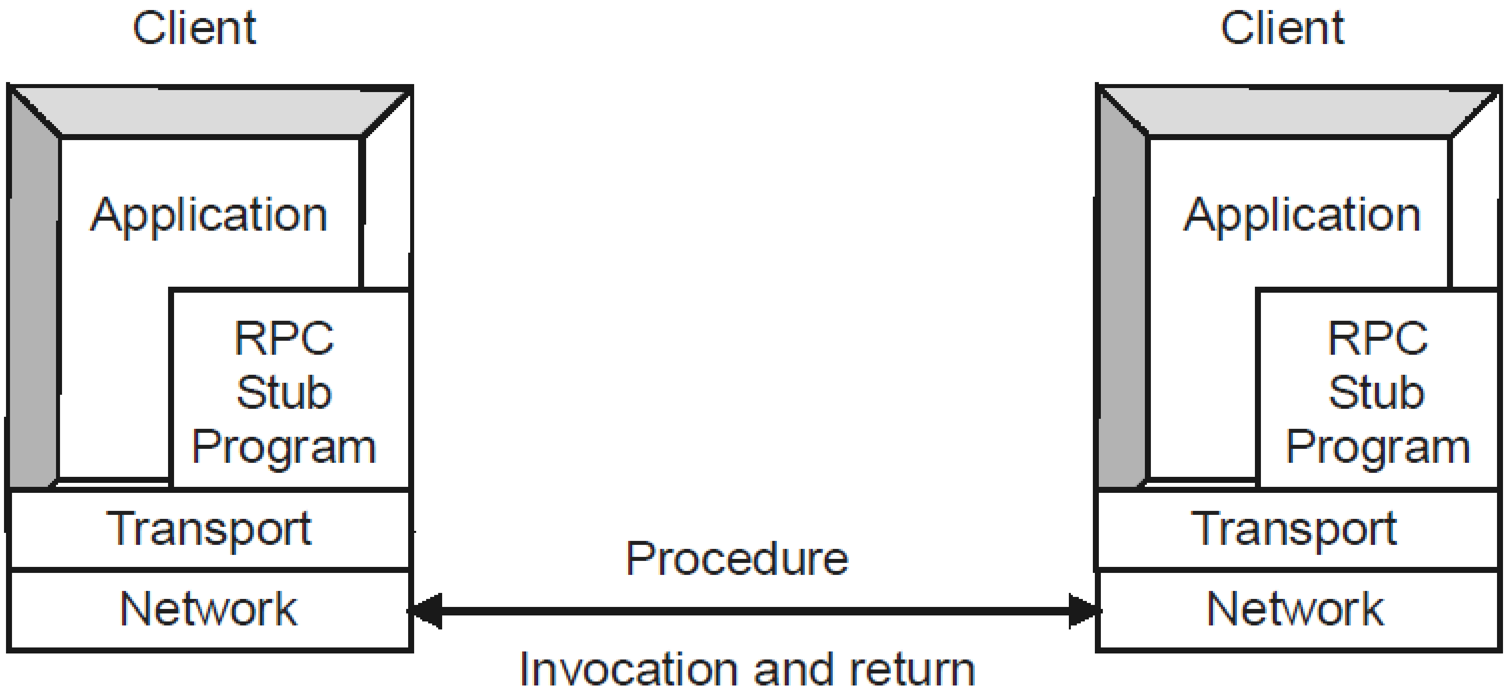


Fig. 5.8: *General Architecture*

- When the client invokes the remote procedure RPC calls the appropriate stub, passes the parameters to it, which are then provided, to remote procedure.
- This stub locates the port on the server and **marshalling** involves packaging the parameter into a form, which may be transmitted over network.
- The stub then transmits a message to server using message passing.
- Now the message sent by the host is received at the client side with the help of similar type of stub.

# 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 windowing services handle placement and manipulation of the application window.***
- So 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 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.

# 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.

# 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.

# Other Remote Services

- ***Applications can be invoked from the client to execute remotely on a server***.
- **Backup services** are an example of services that might be remotely invoked from a client workstation.
- Business functions such as downloading data from a host or checking a list of stock prices might also be invoked locally to run remotely.
- Software is provided by the NOS to run on the client workstation to initiate these remote applications.
- Mobile computing is increasingly being used to remain functional while out of the office.
- Applications can be built to operate effectively from the office LAN or the remote laptop.
- The IPC protocol of choice for mobile access is TCP/IP based.

# Utility Services

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

# Dynamic Data Exchange (DDE)

- DDE is described as a conversation between two applications, a client application and a server application.
- It **sends messages** between applications that share data and **uses shared memory** to exchange data between applications.
- **DDE is a feature of some operating systems (like Windows 98, OS/2) presentation manager that enable users to pass data between applications to application.**
- For eg:, if an application wants to connect a Microsoft Excel spreadsheet with Microsoft Word for windows report in such a way that changes to the spreadsheet are reflected automatically in the report, in that case Microsoft Word for windows is the client and Microsoft Excel is the server.

# Uses for Windows Dynamic Data Exchange

- DDE is most appropriate for data exchanges that **do not require ongoing user interaction.**
- An application provides a method for the user to establish the link between the applications exchanging data.
- Once that link is established, the applications exchange data without further user involvement.
- DDE can be used to implement a broad range of application features:
- **Linking to real-time data,** such as to stock market updates, scientific instruments, or process control.
- **Creating compound documents**, such as a word processing document that includes a chart produced by a graphics application. Using DDE, the chart will change when the source data is changed, while the rest of the document remains the same.

- Two applications participating in DDE are said to be engaged in a **DDE conversation**.
- The application that initiates the conversation is the **DDE client application**; the application that responds to the client is the **DDE server application**.
- An application can engage in several conversations at the same time, acting as the client in some and as the server in others.

- The DDE protocol identifies the units of data passed between the client and server with a three-level hierarchy of application, topic, and item names.
- Each DDE conversation is uniquely defined by the application name and topic.
- At the beginning of a DDE conversation, the client and server determine the application name and topic.
- The application name is usually the name of the server application. For eg, when Excel acts as the server in a conversation, the application name is Excel.
- For applications that operate on file-based documents, the topic is usually a filename.
- A **DDE data item** is information related to the conversation topic exchanged between the applications.

- For eg, if the Word document is to receive data automatically from a range named IBM in a Excel worksheet, named STOCKS.XLS then STOCKS.XLS is the topic and IBM is the item.
- A simplest way to set up a DDE link is to copy a block of data from the server application to the clipboard, activate the client application, move the insertion point to the location in the receiving document where you want the information to go, and then use a Paste Link command.

# Dynamic Data Exchange Message Flow

- A typical DDE conversation consists of the following events:

1. The client application initiates the conversation, and the server application responds.
2. The applications exchange data by any or all of the following methods:
   - The server application sends data to the client at the client's request.
   - The client application sends unsolicited data to the server application.
   - The client application requests the server application to notify the client whenever a data item changes (warm data link).
   - The client application requests the server application to send data whenever the data changes (hot data link).
   - The server application carries out a command at the client's request.
3. Either the client or server application terminates the conversation.

- Once a DDE conversation has begun, the client can establish one or more permanent data links with the server.
- There are two kinds of permanent DDE data links: warm and hot.
- In a warm data link, the server notifies the client that the value of the data item has changed, but the server does not send the data value to the client until the client requests it.
- In a hot data link, the server immediately sends the changed data value to the client.

- A DDE link may be **automatic or manual**.
- An automatic link is refreshed whenever the source data changes, provided both the client and server applications are running.
- A manual link is refreshed only when you issue a command in the client application.

# 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.

- 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.

- OLE 2.0 extends OLE capabilities to enable **a group of data to be defined as an object and saved into a database.**
- This object can then be dragged and dropped into other applications and edited without the need to switch back to the application which created it.
- This provides a more seamless interface for the user.

- 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.

- In OLE 1.x, double-clicking a Lotus 1-2-3 for Windows spreadsheet embedded in a Microsoft Word for Windows document launches 1-2-3 and opens the document in a 1-2-3 window.
- Under OLE 2.0, the active window (Word's) menu and toolbar change to that of 1-2-3.
- The user deals only with the object, with no need to be aware of the multiple software being loaded.

# Common Object Request Broker Architecture (CORBA)

- CORBA is a standard defined by the OMG (**Object Management Group**). It describes an architecture, interfaces, and protocols that **distributed objects can use to interact with each othe**r.
- OLE focuses on data sharing between applications on a single desktop.
- **CORBA addresses cross-platform data transfer and the process of moving objects over networks.**
- CORBA support enables Windows and UNIX clients to share objects.
- Eg: A word processor operating on a Windows desktop can include graphics generated from a UNIX workstation.

# CORBA – Common Object Request Broker

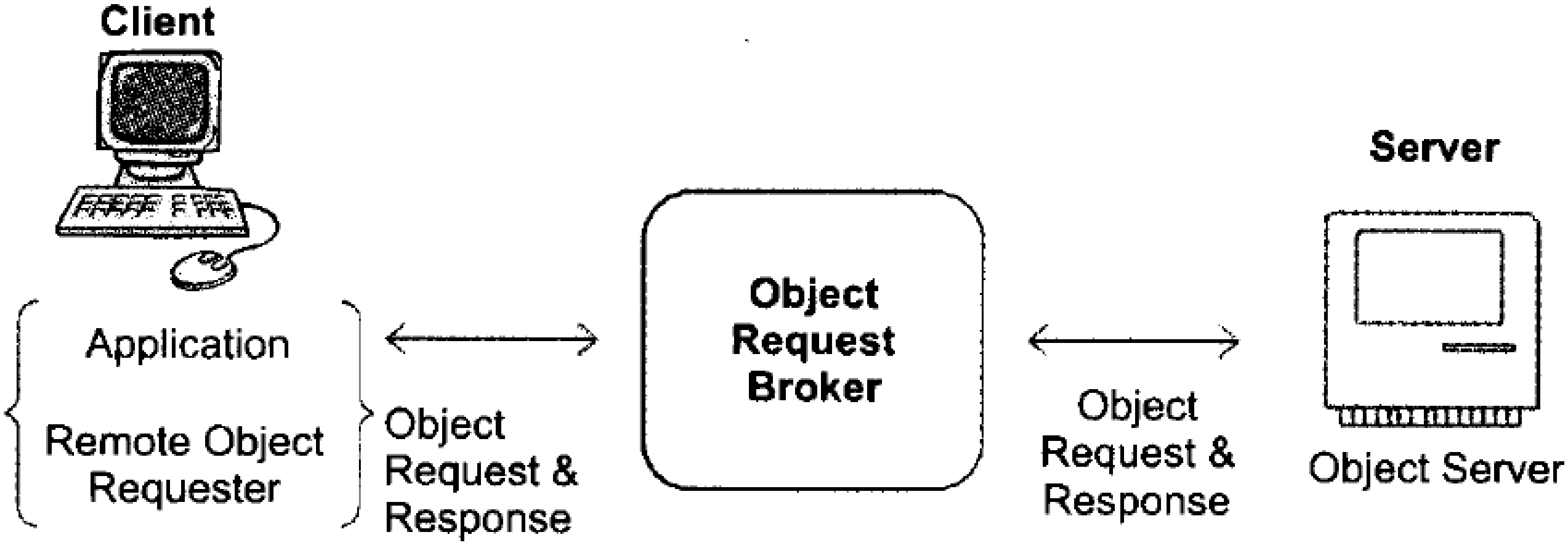
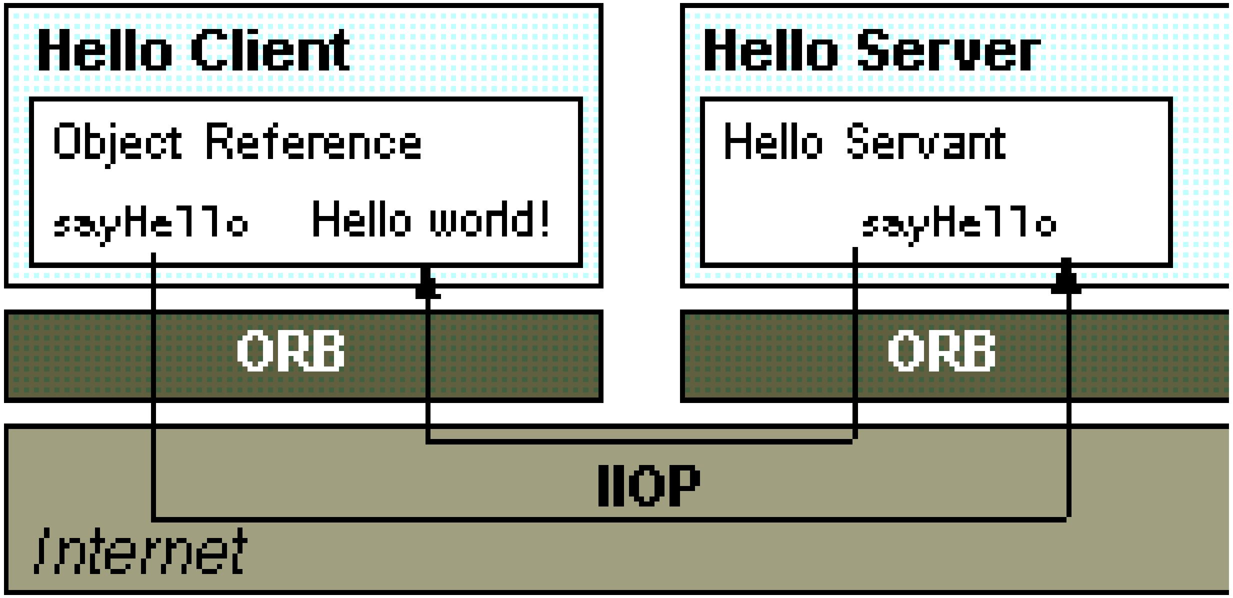


Fig. 5.9: Object Request Broker

- CORBA is a standard for distributing objects across networks so that operations on those objects can be called remotely.
- CORBA is not associated with a particular programming language, and any language with a CORBA binding can be used to call and implement CORBA objects.
- Part of the CORBA standard is the **Interface Definition Language (IDL),** which is an implementation-independent language for describing the interfaces of remote objects.

- CORBA includes the following components:
- **Object Request Broker (ORB)** : responsible for all interactions between remote objects and the applications that use them. ORB handles the communication, marshaling, and unmarshaling of parameters so that the parameter handling is transparent for a CORBA server and client applications.
- **CORBA server :** The CORBA server creates CORBA objects and initializes them with an ORB. The server places references to the CORBA objects inside a naming service so that clients can access them.
- **CORBA Request node :** The CORBARequest node acts as a CORBA client.

- **Naming service :** The naming service holds references to CORBA objects.
- **Interface Definition Language (IDL) :** which is how CORBA interfaces are defined.
- **Portable Object Adaptor (POA):** is responsible for object activation/deactivation, mapping object ids to actual object implementations.
- **Internet Inter-ORB Protocol (IIOP) :** CORBA objects communicate with each other over the Internet via the IIOP, which is **TCP-based** but uses no fixed port number.



- Fig: A one-method distributed object shared between a CORBA client and server.

- Any relationship between distributed objects has two sides: the client and the server.
- The server provides a remote interface, and the client calls a remote interface.

- Between the ORBs, communication proceeds by means of a shared protocol, **IIOP-the Internet Inter-ORB Protocol.**
- **IIOP**, which is based on the standard TCP/IP internet protocol and works across the Internet.
- It defines **how CORBA-compliant ORBs pass information back and forth**.
- Like CORBA and IDL, the IIOP standard is defined by OMG, the Object Management Group.
- **IIOP allows clients using a CORBA product from one vendor to communicate with objects using a CORBA product from another vendor thus permitting interoperability**, which is one of the goals of the CORBA standard**.**

- CORBA-compliant ORBs provides a number of optional services like –
- looking up objects by name
- maintaining persistent objects
- supporting transaction processing
- enabling messaging
- many other abilities useful in today's distributed, multi-tiered computing environments.

# Interface Definition Language (IDL)

- The services that an object provides are given by its interface.
- Interfaces are defined in OMG's Interface Definition Language (IDL).
- IDL is independent of any programming language.
- Mappings from IDL to specific programming languages are defined as part of the CORBA specification.
- Mappings for C, C++, Smalltalk, Ada, COBOL, and Java have been approved by OMG.
- **In IDL, you declare only the names and types for interfaces, data members, methods, method arguments etc.**
- You do not include the method implementations.

- The method implementations are created in implementation language you choose after you’ve used an IDL compiler (*idlj* is the IDL compiler for Java) to convert your IDL interface to your target language.

# Object Request Broker (ORB)

- The core of the CORBA architecture is the ORB.
- Each machine involved in a CORBA application must have an ORB running in order for processes on that machine to interact with CORBA objects running in remote processes.
- **Object clients and servers make requests through their ORBs and the remote ORB locates the appropriate object and passes back the object reference to the requestor.**
- The ORB provides the communication infrastructure needed to identify and locate objects, handles connection management, etc.
- The ORBs communicate with each other via the IIOP.

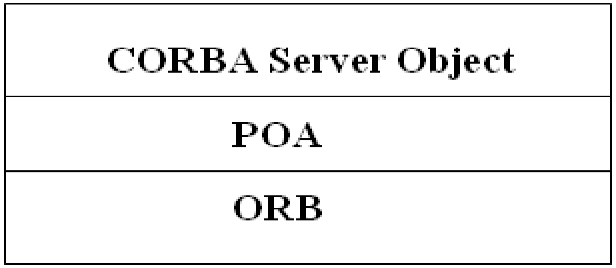
## Naming Service

- Defines how CORBA objects can be looked up by a name. It is a ***Common Object Service (COS)*** and allows an object to be published using a symbolic name and allows clients to obtain references to the object using a standard API.
- The CORBA naming service provides a naming structure for remote objects.

## IIOP

- The CORBA standard includes specifications for inter-ORB communication protocols that transmit object requests between various ORBs running on the network.

# Portable Object Adaptor (POA)



CORBA Server Object

POA

ORB

- The POA connects the server object implementation to the ORB.  It extends the functionality of the ORB and some its services include:
- activation and deactivation of the object implementations
- generation and management of object references
- mapping of object references to their implementations
- dispatching of client requests to server objects through a skeleton

# Server Functionality in Detail

- The functions provided by the server in a NOS environment:
  - **Request Processing**
  - **File Services**
  - **Fax/Print/Image Services**
  - **Database Services**
  - **Communications Services**
  - **Security Services**

# 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.

# 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.
- **All requests are mapped into the virtual pool of resources and redirected as necessary to the appropriate local or remote server.**
- The file services provide this support at the remote server processor.
- In the typical implementation, software, shared data, databases, and backups are stored on disk, tape, and optical storage devices that are managed by the file server.

- To minimize the effort and effect of installation and maintenance of software, software should be loaded from the server for execution on the client.
- **New versions can be updated on the server** and made immediately available to all users.
- **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.

- 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.
- 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.

# 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.

# 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.
- 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.

# Database trends.

- **Flat Files: Sorting Physical Records -** These original implementations physically stored data columns and records according to the user view.
- **Hierarchical Databases: Adjacent Storage of Related Record Types -** could store related record types physically or logically next to each other.
- The developer explicitly references the index to get to the data of interest.
- Disadvantage is that only applications that access data according to its physical storage sequence benefit from locality of reference.

- **Relational Databases: Extracted Indexes and SQL**
- Relational database technology provides the current data management solution to many of the problems inherent in the flat-file and hierarchical technologies.
- The first relational products were introduced by ADR with Datacom DB and Computer Corporation of America with Model 204.
- Used extracted indexes to provide direct access to stored data without navigating the database or sorting flat files.

- The primary design objective behind SQL was to **provide a data access language** that could be shown mathematically to manipulate the desired data correctly.
- The secondary objective was to **remove any sense of the physical storage of data from the view of the user.**
- SQL uses extracted indexes to provide direct access to the rows (records) of the tables (files) of interest.
- Each column (field) may be used as part of the search criteria.
- SQL provides a very powerful data access language.
- Its algebra provides all the necessary syntax to define, secure, and access information in an SQL database.

- For the development of client/server applications, products should be implemented with support for the following products and standards:
- ANSI SQL and IBM DB2 standards
- A variety of front-end query products
- C and COBOL SQL precompilers
- Support for and compatibility with server NOS: NetWare, OS/2 (LAN Manager, LAN Server), Windows NT
- Support for client Operating Systems: DOS, Windows, OS/2, Windows NT, Mac System 7, or UNIX

- The following DBMS features must be included in the database engine:
  - Performance optimization tools
  - Dynamic transaction backout
  - Roll back from, roll forward to last backup
  - Audit file recovery
  - Automatic error detection and recovery
  - File reclamation and repair tools
  - Support for mirrored databases
  - Capability to split database between physical disk drives
  - Remote distributed database management features
  - Maintenance of accurate and duplicate audit files on any LAN node

- In the client/server implementation, you should **offload database processing to the server**.
- The database engine should provide **support for stored procedures or triggers** that run on the server.
- The client/server model implies that there will be **multiple concurrent user access**.
- The database engine must be able to manage this access without requiring every developer to write well-behaved applications.
- The following features must be part of the database engine:
  - **Locking mechanisms to guarantee data integrity**
  - **Deadlock detection and prevention**
  - **Multithreaded application processing**
  - **User access to multiple databases on multiple servers**

# OODBMS

- Object-oriented database management systems provide support for complex data structures: such as compound documents, CASE entity relationship models, financial models, and CAD/CAM drawings.

# 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.

# Security Services

- Client/server applications require similar security services to those provided by host environments.
- **Every user should be required to log in with a user ID and password.**
- If passwords might become visible to unauthorized users, the security server should insist that passwords be changed regularly.
- The enterprise on the desk implies that a single logon ID and logon sequence is used to gain the authority once to access all information and process for the user has a need and right of access.

- Because data may be stored in a less physically secure area, **the option should exist to store data in an encrypted form.**
- A combination of the user ID and password should be required to decrypt the data.
- New options, such **as floppyless workstations with integrated data encryption standard (DES) coprocessors**, are available.
- 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.

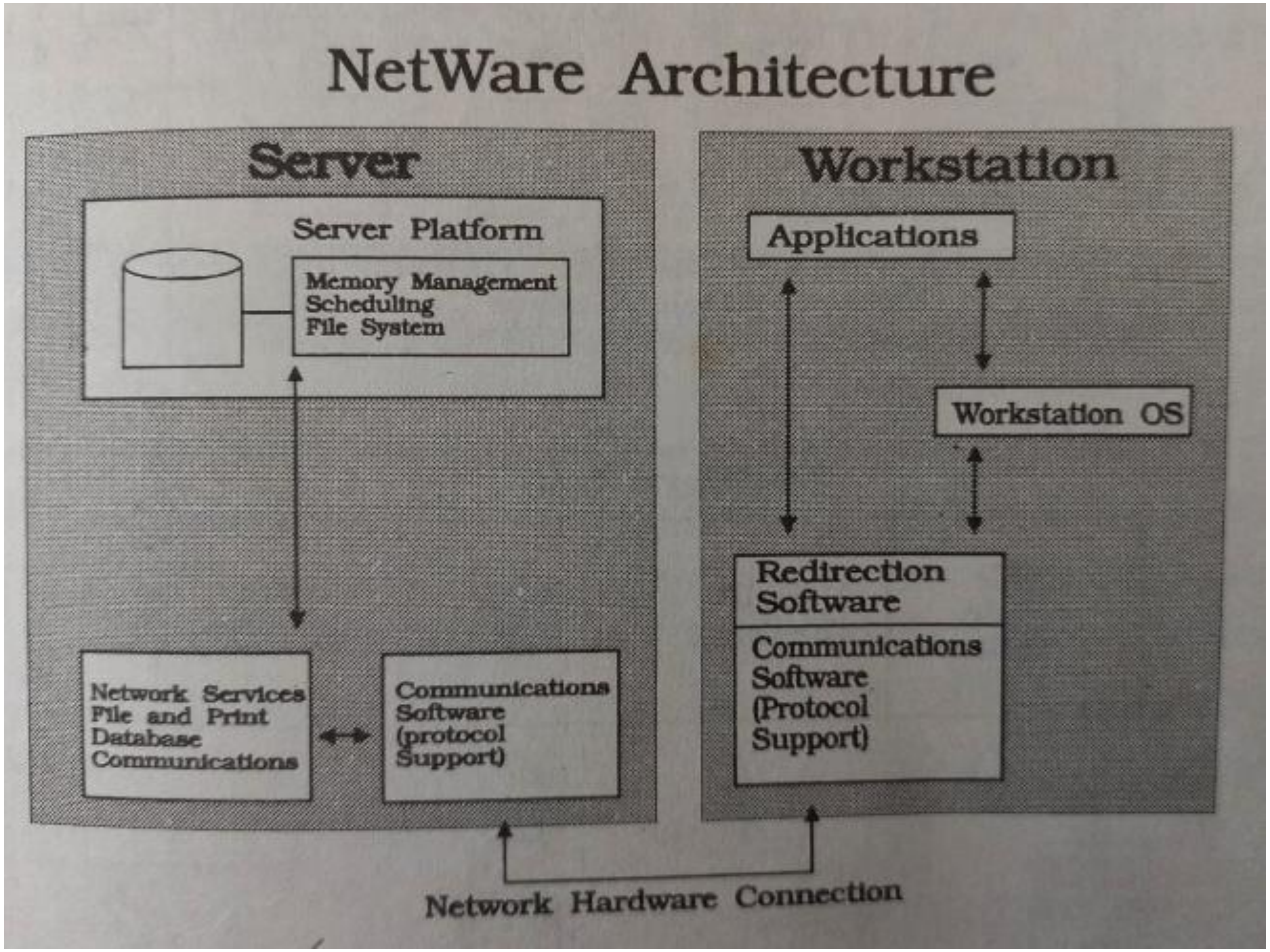
# The Network Operating System (NOS)

- The network operating system (NOS) provides the services not available from the client OS.
  - Novell NetWare
  - LAN Manager
  - IBM LAN Server
  - Banyan VINES
  - PC Network File Services (NFS)

# 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.
- NetWare provides the premier LAN environment for **file and printer resource sharing**.

- It had 62 percent of the market share in 1993.
- It is widely installed as the **standard product** in many organizations.
- ***NetWare is the original LAN NOS for the PC world.***
- It incorporates many of the ease-of-use features required for sharing printers, data, software, and communications lines.



- Figure shows the major components of the NetWare architecture, illustrating client and server functions.

- 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.
- Figure 4.5 outlines the **NetWare open architecture.**

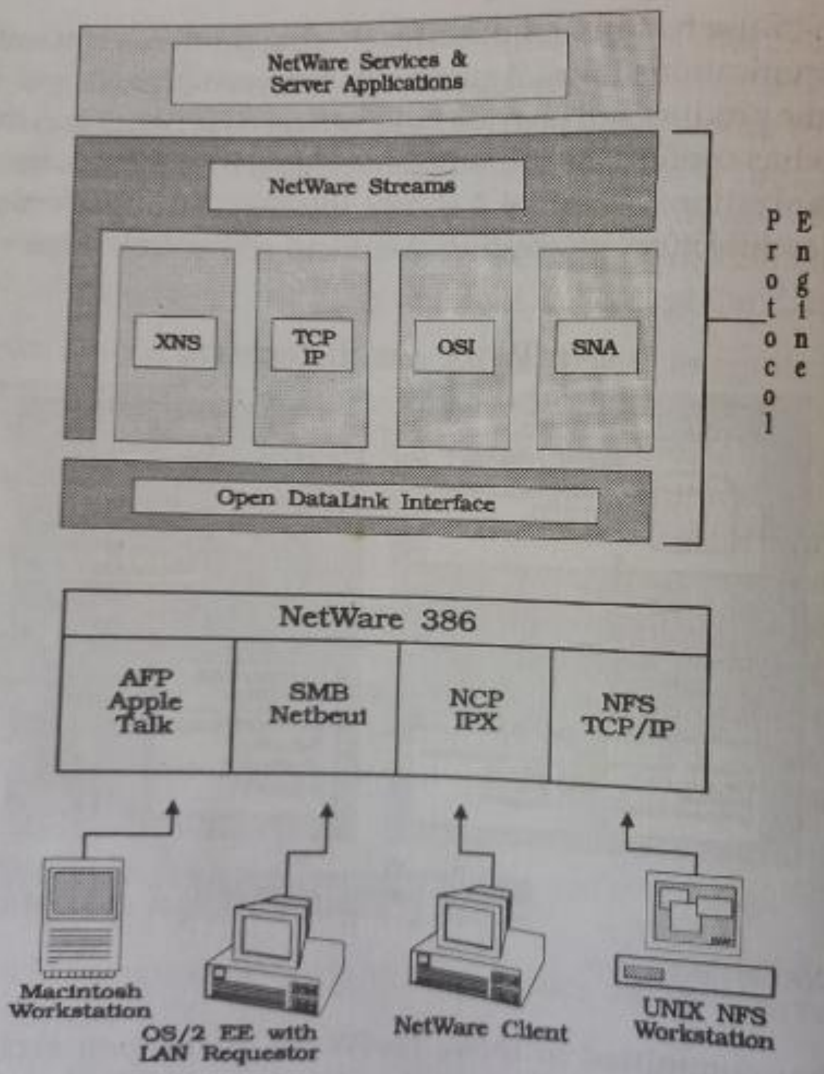


*Figure 4.5. NetWare open services.*

- NCP IPX- **Internetwork Packet Exchange** using Novell Netware OS
- NCP - Netware Core Protocol
- SMB – Server Message Block protocol
- Netbeui – NetBIOS Extended User Interface.
- XNS – Xerox Network Services.

- Client workstations can use Mac System 7, OS/2, DOS, Windows, Windows NT, NetWare, or UNIX NFS operating environments.
- OS/2, Windows NT, and UNIX servers may be installed on the same LAN as NetWare servers to provide support for products that require these platforms.
- Novell's goal is to provide **NetWare services totally independent of network media, network transport protocols, client/server protocols, and server and client operating systems, at each layer of network design**.

- Adv:
  - high performance
  - low resource requirements
  - relative ease of use
- 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.

# LAN Manager

- **LAN Manager** and its IBM derivative, **LAN Server**, are the standard products for use in client/server implementations using OS/2 as the server operating system.
- **LAN Manager/X** is the standard product for client/server implementations using UNIX System V as the server operating system.
- Microsoft released its **Advanced Server** product with Windows NT in the third quarter of 1993.

- During 1994, it will be enhanced with support for the **Microsoft network management services** and **Banyan's Enterprise Network Services (ENS)**.
- Advanced Server is the natural migration path for existing Microsoft LAN Manager and IBM LAN Server customers.
- LAN Manager and Advanced Server provide client support for DOS, Windows, Windows NT, OS/2, and Mac System 7.
- Client workstations can access data from both NetWare and LAN Manager servers at the same time.

- LAN Manager supports **NetBIOS** and **Named Pipes LAN communications** between clients and OS/2 servers.
- **Redirection services** are provided to map files and printers from remote workstations for client use.
- ***Advanced Server also supports TCP/IP communication.***
- Microsoft has added **TCP/IP support** to LAN Manager 2.1 and Advanced Server along with NetView and Simple Network Management Protocol (SNMP) agents.

- **Advanced Server** provides *integrated support for peer-to-peer processing and client/server applications.*
- Existing support for Windows NT, OS/2, UNIX, and Mac System 7 clients lets application, database, and communication servers run on the same machine as the file and print server.
- This feature is attractive in small LANs.

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

# 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 (Simple Network Architecture) 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(Advanced Peer-to-Peer Networking).

- IBM has priced LAN Server very attractively so that if OS/2 clients are being used, LAN Server is a low-cost option for small LANs.
- LAN Server supports DOS, Windows, and OS/2 clients.
- No support has been announced for Mac System 7, although it is possible to interconnect AppleTalk and LAN Server LANs to share data files and communication services.

# 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.
- **StreetTalk names** follow a three-level hierarchical format: **Item@Group@Organization.**
- For example, a user can be identified as Psmith@Cerritos@Tnet.
- All network objects are stored in a distributed database that can be accessed globally.

- Novell's NDS is similar to StreetTalk in functionality.
- StreetTalk is less flexible but less complex to manage.
- One advantage of StreetTalk has over NDS is that StreetTalk **objects can have unlimited attributes available for selection**.
- To locate a printer with certain attributes, the command: "Locate a color laser printer with A4 forms on the 7th floor of Cerritos" finds and uses the printer with the desired characteristics.
- **VINES V5.5 offers ISDN and TI support for server-to-server communications over a WAN**, as well as integration of DOS, Windows, OS/2, and Mac clients.
- VINES does not support NFS clients.

# 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.

- With the increasing use of UNIX servers for application and database services, there is an increasing realization that PC NFS may be all that is required for NOS support for many workgroups.
- This can be a low-cost and low-maintenance option because the UNIX server is easily visible from a remote location.

# Available Platforms

- Client/server computing requires that **LAN and WAN topologies be in place** to provide the necessary internetworking for shared applications and data.

- **Workstations in LAN Configuration**
- **LAN-to-LAN/WAN Configuration**
- **LAN-to-Host Configuration**
- **Enterprise-Wide**
  - **OLTP on a LAN**
  - **OLTP with UNIX**

# Workstations in LAN Configuration

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

# LAN-to-LAN/WAN Configuration

- **Routers and communication servers** will be used to provide communication services between LANs and into the WAN.
- In the client/server model, these connections will be provided transparently by the SDE (Systems Development Environment) tools.
- There are significant performance implications if the traffic volumes are large.
- IBM's LU6.2 (Logical Unit type 6.2 protocol) implementation in **APPC (Advanced Program-to-Program Communication)** and **TCP/IP** provides the best support for high-volume, LAN-to-LAN/WAN communications.

- DEC's implementation of **DECnet (Digital Equipment Corporation Networking architecture)** always has provided excellent LAN-to-WAN connectivity.
- Integrated support for TCP/IP, LU6.2, and IPX (Internetwork Packet Exchange) provides a solid platform for client/server LAN-to-WAN implementation within DECnet.
- Novell 4.x provides support for TCP/IP as both the LAN and WAN protocol.
- Internetworking also is supported between IPX and TCP/IP.

# LAN-to-Host Configuration

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

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

# Enterprise-Wide

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

- Wherever data resides or is used, the location must be transparent to the user and the developer.
- Data should be stored where it best meets the business need.

# OLTP on a LAN

- Online Transaction Processing applications are found in such industries as insurance, finance, government, and sales - all of which process large numbers of transactions.
- The systems have high requirements for availability, data integrity, performance, concurrent access, growth potential, security, and manageability

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

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

# OLTP with UNIX

- UNIX has added many of the features found in other commercial operating systems such as VMS(Virtual Memory System) and MVS(Multiple Virtual Storage).
- There are now several offerings for OLTP with UNIX.
- Database services will be provided by a combination of AIX and MVS servers.
- (IBM's Advanced Interactive eXecutive, or AIX, is a series of proprietary UNIX-based operating systems built and sold by IBM).

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

# The Server Operating System

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

# NetWare

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

# 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.

# Windows NT

- With the release of Windows NT (New Technology) in September of 1993, Microsoft staked its unique position with a server operating system.
- 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.

# MVS

- IBM provides MVS as a platform for large applications.
- 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 (Systems Application Architecture).**
- The objective of SAA is to provide all services on all IBM platforms in a compatible way.
- IBM provides support for the LAN Server running natively under MVS.
- MVS provides a powerful database server using DB2 and LU6.2.

# OPENVMS

- NetWare supports the use of OPENVMS servers for file services.
- DEC provides its own server interface using a LAN Manager derivative product called Pathworks.

- 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.

- VAX OPENVMS support for database products such as RDB, Sybase, Ingres, and Oracle enables this platform to execute effectively as a database server for client/server applications.