

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

DISEÑO DE REDES UTILIZANDO LA ARQUITECTURA CLIENTE/SERVIDOR

TESIS

PRESENTADA A LA JUNTA DIRECTIVA
DE LA FACULTAD DE INGENIERÍA
POR

ROLANDO HAROLDO ALONSO ORDOÑEZ
PREVIO A OPTAR EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

Guatemala, Agosto de 1,996.

[Faint, illegible text or signature]

08
7(3757)
C.4

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de tesis titulado:

DISEÑO DE REDES UTILIZANDO LA ARQUITECTURA CLIENTE/SERVIDOR

tema que me fuera asignado por la Coordinación de la carrera de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería.

Rolando Haroldo Alonso Ordoñez

PROPIEDAD DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
Biblioteca Central

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

MIEMBROS DE LA JUNTA DIRECTIVA

DECANO	Ing. Julio Ismael González Podszueck
VOCAL 1o.	Ing. Miguel Angel Sánchez Guerra
VOCAL 2o.	Ing. Jack Douglas Ibarra Solórzano
VOCAL 3o.	Ing. Juan Adolfo Echeverria Méndez
VOCAL 4o.	Br. Fernando Waldemar De León Contreras
VOCAL 5o.	Br. Pedro Ignacio Escalante pastor
SECRETARIO	Ing. Francisco Javier González López

TRIBUNAL QUE PRACTICO EL EXAMEN
GENERAL PRIVADO

DECANO	Ing. Jorge Mario Morales González
EXAMINADOR	Ing. Armando Amado
EXAMINADOR	Lic. David Alvarez
EXAMINADOR	Ing. Frisly Fuentes
SECRETARIO	Ing. Edgar Bravatti Castro

Guatemala, Junio de 1996

Ingeniero
Calixto Raúl Monzón P.
Coordinador de la carrera,
Ingeniería en Ciencias y Sistemas.

Ing. Monzón

Por este medio me permito informarle que revisé el trabajo de tesis titulado **DISEÑO DE REDES UTILIZANDO LA ARQUITECTURA CLIENTE/SERVIDOR**, elaborado por el estudiante **ROLANDO HAROLDO ALONSO ORDOÑEZ** y, a mi juicio, la misma cumple con los objetivos propuestos para su desarrollo.

Sin otro particular, me suscribo de usted.

Atentamente,



ING. FRANCISCO GUEVARA CASTILLO

Guatemala, Julio de 1996

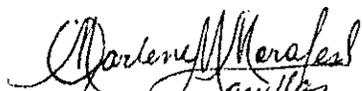
Ingeniero
Calixto Raúl Monzón P.
Coordinador de la carrera
Ingeniería en Ciencias y Sistemas.

Respetable Ing. Monzón

Por este medio me permito informarle que revisé el trabajo de tesis titulado **DISEÑO DE REDES UTILIZANDO LA ARQUITECTURA CLIENTE/SERVIDOR**, elaborado por el estudiante **ROLANDO HAROLDO ALONSO ORDOÑEZ** y, a mi juicio, la misma cumple con los objetivos propuestos para su desarrollo.

Sin otro particular, me suscribo de usted.

Atentamente,


Ing. Marlene Morales Masella

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

Escuelas de Ingeniería Civil, Ingeniería
Mecánica Industrial, Ingeniería Química,
Ingeniería Mecánica Eléctrica, Técnica
y Regional de Post-grado de Ingeniería
Sanitaria.

Ciudad Universitaria, zona 12
Guatemala, Centroamérica

Guatemala, Julio de 1996

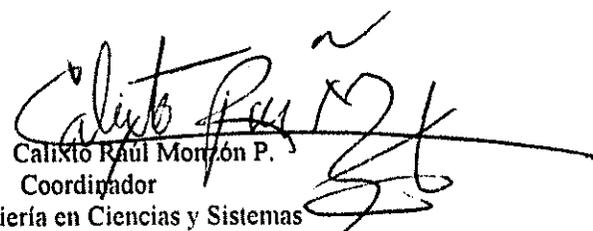
Ingeniero
Julio Ismael Gonzáles Podzueck
Decano
Facultada Ingeniería

Señor Decano :

Me dirijo a usted para informarle que después de conocer el dictamen del asesor del trabajo de tesis titulado **DISEÑO DE REDES UTILIZANDO LA ARQUITECTURA CLIENTE/SERVIDOR**, elaborado por el estudiante **ROLANDO HAROLDO ALONSO ORDOÑEZ**, procedo a la autorización del mismo.

Sin otro particular, me suscribo de usted.

Atentamente,


Ing. Calixto Raúl Morán P.
Coordinador
Ingeniería en Ciencias y Sistemas



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

Escuelas de Ingeniería Civil, Ingeniería
Mecánica Industrial, Ingeniería Química,
Ingeniería Mecánica Eléctrica, Técnica
y Regional de Post-grado de Ingeniería
Sanitaria.

Ciudad Universitaria, zona 12
Guatemala, Centroamérica

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la autorización por parte del Coordinador de la Carrera de Ingeniería en Ciencias y Sistemas, al trabajo de tesis titulado "DISEÑO DE REDES UTILIZANDO LA ARQUITECTURA CLIENTE/SERVIDOR", presentado por el estudiante universitario ROLANDO HAROLDO ALONSO ORDOÑEZ, procede a la autorización para la impresión de la misma.

IMPRIMASE:

Ing. Julio Ismael González Podszueck

DECANO



Guatemala, Julio de 1996

DEDICATORIA

A DIOS : Por permitirme terminar mis estudios.

A mis padres : Antonieta de Alonso y Arnoldo Alonso, como un tributo por todo el sacrificio y amor que han hecho a lo largo de mi vida.

A mis Hermanos : Ricardo, Alex y Claudia.

A mi Abuelita: Patrocinia viuda de Ordoñez

A Patty : Por su amor y apoyo en todo momento, es muy especial contar contigo.

AGRADECIMIENTO

Un agradecimiento muy especial al Ing. Francisco Guevara por su valiosa colaboración para la realización de esta tesis.

INDICE

CONTENIDO	No. Página
1. INTRODUCCIÓN A LOS SISTEMAS DISTRIBUIDOS Y REDES DE COMPUTACIÓN	1
1.1. EVOLUCIÓN HACIA LA ARQUITECTURA CLIENTE SERVIDOR	1
1.1.1 PROCESAMIENTO BASADO EN UN COMPUTADOR CENTRAL	1
1.1.2 PROCESAMIENTO MAESTRO ESCLAVO	2
1.1.3 PROCESAMIENTO CLIENTE/SERVIDOR	3
1.1.4 PROCESAMIENTO PUNTO A PUNTO	6
1.1.5 VENTAJAS DE LA COMPUTACIÓN CLIENTE/SERVIDOR	7
1.2 ESTÁNDARES Y SISTEMAS ABIERTOS	8
1.2.1 SISTEMAS ABIERTOS	9
1.2.2 BONDADDES Y ESTÁNDARES PROPIETARIOS	10
1.2.3 BENEFICIARIOS DE LOS SISTEMAS ABIERTOS	11
1.3 AMBIENTE DE COMPUTACIÓN DISTRIBUIDO (DCE)	13
1.3.1 ANTECEDENTES	13
1.3.2 ARQUITECTURA DCE	14
2. DESARROLLO DE SISTEMAS DE INFORMACIÓN DISTRIBUIDO	19
2.1. MODELOS DISTRIBUIDOS	20
2.1.1 AMBIENTE MULTINIVEL	20
2.1.2 EL PROCESAMIENTO COOPERATIVO CLIENTE/SERVIDOR	22
2.1.3 TÉCNICAS DE PROCESAMIENTO COOPERATIVAS	28
2.2 SISTEMAS DE IMAGEN ÚNICO	30
2.3 PRESENTACIÓN DISTRIBUIDA	30
2.4 PROCESAMIENTO DISTRIBUIDO	32
2.5 INFORMACIÓN DISTRIBUIDA	35
2.6 DISTRIBUCIÓN Y ADMINISTRACIÓN DEL SOFTWARE	39
3. ESPECIALIZACIÓN DEL CLIENTE Y DEL SERVIDOR EN AMBIENTES DISTRIBUIDOS	40
3.1 CLIENTES	40
3.1.1 EL ROL Y LAS FUNCIONES DEL CLIENTE	40
3.1.2 ADMINISTRACIÓN DE LA PRESENTACIÓN	41
3.1.3 CARACTERÍSTICAS GUI	43
3.1.4 HERRAMIENTAS COMPATIBLES	47
3.2 SERVIDORES	49
3.2.1 FUNCIONES	49
3.2.2 ARQUITECTURA DEL HARDWARE DEL SERVIDOR	54
3.2.3 MULTIPROCESAMIENTO SIMÉTRICO	56

4.	REDES Y COMUNICACIONES	60
4.1	SISTEMAS DE COMUNICACIÓN	60
4.1.1	COMUNICACIONES Y DISTRIBUCIÓN	60
4.1.2	FUNCIONES DEL SISTEMA DE COMUNICACIÓN	62
4.1.3	CAPAS, PROTOCOLOS E INTERFACES	64
4.1.4	MODELO DE REFERENCIA ISO	66
4.1.5	CONECTIVIDAD DE LOS COMPONENTES CLIENTE/SERVIDOR	70
4.1.6	COMUNICACIÓN Y SINCRONIZACIÓN	70
4.1.7	REDES DE TRABAJO	74
	A. CLASIFICACIÓN	74
	B. TOPOLOGÍA	75
	C. TÉCNICAS DE CONMUTACIÓN EN REDES	78
4.2	REDES DE ÁREA LOCAL	79
4.2.1	CONCEPTOS DE REDES DE ÁREA LOCAL	79
4.2.2	TRANSMISIÓN Y MÉTODOS DE CONTROL DE ACCESO	86
4.2.3	ARQUITECTURA IEEE 802.2	86
5.	PROTOCOLOS UTILIZADOS EN LANS	92
5.1	NetBIOS/NetBEUI	92
5.1.1	NetBEUI	93
5.1.2	NetBIOS	93
5.2	ESPECIFICACION DE INTERFACE CONTROLADOR DE RED	95
5.3	TUBERIAS (PIPES)	97
5.4	TUBERÍAS CON NOMBRE (NAMED PIPES)	98
5.5	INTERCAMBIO DE PAQUETES EN REDES INTERCONECTADAS	99
5.6	TCP/IP	101
5.7	IMPLEMENTACIONES LAN	106
5.8	CONECTIVIDAD INTER-LAN	111
6	APLICACIÓN : DISEÑO DE UNA RED CLIENTE/SERVIDOR	113
7	CONCLUSIONES	118
8	RECOMENDACIONES	119
8.	BIBLIOGRAFÍA	120

INDICE DE GRAFICAS

Gráfica	No Página
Figura 1.1 Ambiente de procesamiento basado en un Host	1
Figura 1.2 Ambiente de procesamiento Maestro-Esclavo	2
Figura 1.3 Ambiente de procesamiento Cliente/Servidor	3
Figura 1.4 Ambiente de procesamiento de dispositivos compartidos	4
Figura 1.5 Ambiente de procesamiento punto a punto	6
Figura 1.6 Modelo Cliente/Servidor DCE	17
Figura 2.1 Los modelos cliente/servidor	19
Figura 2.2 Arquitectura de tres niveles	21
Figura 2.3 Arquitectura de tres niveles extendida	22
Figura 2.4 Componentes de una aplicación típica	24
Figura 2.5 Procesamiento cooperativo y distribuido Cliente/Servidor	25
Figura 2.6 Arquitectura de tres niveles con acceso de datos heterogéneo	27
Figura 2.7 Procesamiento Cliente/Servidor	28
Figura 2.8 Presentación distribuida	31
Figura 2.9 Procesamiento distribuido	34
Figura 2.10 Conexión Directa	37
Figura 2.11 Conexión Indirecta	38
Figura 3.1 Modelo de ciclo de evento	44
Figura 3.2 Modelo CallBack de eventos	45
Figura 3.3 Varios APIs GUI	47
Figura 3.4 Compatibilidad de herramientas APIs	48
Figura 3.5 Funciones Cliente/Servidor	50
Figura 3.6 Mainframe como Servidor	53
Figura 3.7 Arquitectura RISC RS/6000 superescalar	54
Figura 3.8 Multiprocesamiento de memoria compartida simétrico	58
Figura 4.1 Sistema de comunicación y red en un ambiente distribuido	61
Figura 4.2 Segmentación	62
Figura 4.3 Capas OSI	67
Figura 4.4 Implementación RPC	73
Figura 4.5 Comunicación Punto a Punto	74
Figura 4.6 Redes de comunicación Multipunto	75
Figura 4.7 Interconexión Completa	76
Figura 4.8 Topología Estrella	76
Figura 4.9 Topología Jerárquica o de Árbol	77
Figura 4.10 Topología Bus	77
Figura 4.11 Topología Anillo	78
Figura 4.12 Componentes LAN	82
Figura 4.13 OSI y el Proyecto IEEE 802	87
Figura 4.14 Capas del Proyecto IEEE 802	88
Figura 4.15 Traslado de token bus IEEE 802.4	90
Figura 4.16 Traslado de token ring IEEE 802.5	91
Figura 5.1 Ambiente del protocolo NetBIOS/NETBEUI	92
Figura 5.2 Interface NDIS de Microsoft	95
Figura 5.3 Configuración NDIS	96
Figura 5.4 Pipes y Named Pipes	97
Figura 5.5 Arquitectura TCP/IP	103
Figura 5.6 Red Vines	108
Figura 5.7 Arquitectura Jerárquica MAP/TOP	110
Figura 6.1 Red basada en un Host	114
Figura 6.2 Red basada en un Host	117

GLOSARIO

API: un API (Interface de programación de aplicación) es un conjunto de librerías de rutinas para una GUI específica.

Backlogs : acceso remoto a un computador, que emula una terminal de un sistema centralizado.

Bloque: es un número de caracteres de datos contiguos que forman un mensaje o parte de él.

Bridgets (puentes) : es un dispositivo de redes que proporciona un camino de comunicación entre dos o más segmentos de una red o entre dos o más redes; los puentes proporcionan funciones de filtro leyendo las direcciones en la información enviada, para determinar a que segmento de la red pertenece el paquete de datos. Sin embargo, éstos no pueden proporcionar el mejor camino en la red; esta es función de los ruteadores.

Commitment : cuando un programa modifica de alguna manera la base de datos, tal modificación deberá considerarse en un principio como tentativa, en el sentido de que si algo sale mal después, la modificación se puede anular; cuando la base de datos se modifica, esta es tentativa hasta que sucedan dos hechos : a) se ejecuta una proposición COMMIT (comprometer), la cual convierte en definitivas ("compromete") todas la modificaciones tentativas; o b) se ejecuta una proposición ROLLBACK (retroceso), la cual anula todas la modificaciones tentativas. Una vez comprometida, una modificación no podrá ser anulada.

Conexión orientada : significa que una sesión es configurada entre dos sistemas de comunicación para monitorear el flujo de paquetes de datos.

Datagrama: es otro nombre que se le da a los paquetes que son transferidos con comunicación sin conexión.

Desktop: tecnología computacional de escritorio, computadora personal de escritorio

Disco duro: unidad de almacenamiento de información de grandes capacidades y de acceso rápido.

Enlace (Link): es un circuito físico entre los puntos, o bien un circuito lógico o conceptual entre dos usuarios de una red de conmutación de paquetes u otro tipo de red de comunicaciones, que les permite comunicarse entre sí.

FDM (Frequency Division Multiplex): es un sistema de transmisión simultánea, en el cual el rango disponible de frecuencias de transmisión es dividido en bandas más angostas, cada una utilizada como un canal separado.

Font: tipo de caracter

Gateways (pasarelas): es un dispositivo que actúa como traductor entre dos o más sistemas que no usan el mismo protocolo de comunicación, formato de datos, lenguajes, y/o arquitectura; un gateway es distinto que un bridge, el cual simplemente pasa información entre dos o más sistemas.

Host: computador central, es el computador que realiza todo el procesamiento de un sistema centralizado.

Icono: imagen, figura o gráfica, llamado así en ambientes windows

Kernel : porción del sistema operativo, que se encarga de controlar todas las operaciones que implican procesos.

Llamadas a procedimiento remoto (RPC): un RPC es un mecanismo por el cual un proceso puede ejecutar otro proceso (subrutina) que reside en un sistema diferente, usualmente remoto, posiblemente corriendo en un sistema operativo diferente.

Mainframe: computador central, computador que realiza todo el procesamiento de un sistema centralizado.

Modem: dispositivo que modula y desmodula señales transmitidas sobre las líneas de comunicación. La sección moduladora es usada para transmisión y la desmoduladora para la recepción. El trabajo del modem es convertir las señales digitales en analógicas y viceversa.

Mouse (ratón): hardware utilizado para facilitar el uso rápido del cursor en la pantalla del computador.

Multiplexor (Multicanalizador de comunicaciones): dispositivo que permite la concentración de líneas que operan a distinta velocidad y con diferentes protocolo, para economizar componentes de comunicaciones.

Pipes (tuberías): los pipes (tuberías) representan un mecanismo orientado a conexión, que pasa información de un proceso a otro.

Pixel: punto en la pantalla que tiene determinados atributos, como color, intensidad.

RollBack (retroceder) : ver commitment.

Router (ruteador): interconecta una red con otras redes, proporciona el control del tráfico y funciones de filtración, cuando más de un camino existe entre dos puntos finales en la red, direcciona los paquetes en el camino más eficiente o económico en la red.

X/Open: organización no lucrativa fundada para desarrollar estándares de interoperabilidad entre sistemas diferentes.

Sin conexión : implica que los paquetes de información son enviados de la fuente al destino sin que antes haya habido establecimiento de la conexión y sin un monitoreo mutuo de intercambio de paquete por ambos. La fuente simplemente asume que el destinatario puede manejar cualquier tipo de información que éste envía. El destino recibe los paquetes y los reordena.

TDM (Time División Multiplex): es una forma de obtener varios canales en un enlace simple, dividiendo el tiempo de uso de dicha vía en varios períodos de tiempo y signando cada uno de los canales, de acuerdo con un criterio preestablecido. En la recepción, cada canal es reensamblado en forma separada.

X Windows System: un sistema de administración de la presentación distribuida desarrollado por MIT para ambientes basados en UNIX

ABREVIATURAS

ADCCP	Advanced Data Communications Control Procedures
AIX	UNIX Advanced Interactive eXecutive
AM	Modulación de amplitud
ANDF	Formato de distribución de software de arquitectura neutral (Architecture Neutral Software Distribution Format)
ANSI	American National Standards Institute
API	Interface de programación de aplicación (Application Programming Interface)
APPC	Comunicación avanzada de programa a programa (Advanced Program to program communications)
AS/400	Application System 400
CAE	Ambientes de aplicación común (Common Aplicacion Enviroment)
CASE	Ingenieria de Software ayudada por computador (computer aides software engineering)
CCITT	Comité Consultor en Telegrafia y Telefonfa Internacional (Consultative Committee on International Telegraphy and Telephony)
CISC	Sistemas de información de control de clientes (Customer Information Control System)
CMA	Arquitectura multitarea concertada (Concert Multithread Architecture)
COS	Corporación para sistemas abiertos (Corporation for Open Systems)
CPU	Unidad central de proceso (central process unit)
CSMA/CA	Acceso múltiple por detección de portación con evasión de colisión
CSMA/CD	Acceso múltiple por detección de portador con detección de colisión
DARPA	Agencia de Proyectos de Investigacion Avanzada de la Defensa
DB2	Database 2 IBM
DBMS	Sistema de Administración de Bases de Datos (Database Management System)
DCE	Ambiente de computación distribuido (Distributed Computing Environment)
DFS	Sistema de archivo distribuido (Distributed Files System)
DML	Lenguaje de Manipulación de los Datos (Data Manipulation Language)
DOD	The Department of Defense
DOE	The Department of Energy
DPI	Puntos por pulgada (dots per inch)
E/S	Entrada/salida (input/output)
FDDI	Cable de fibra óptica
FM	Modulación de frecuencia
FMD	Multiplexación por división de frecuencia
FTMA	Protocolo de administración, acceso y transferencia de archivos
FTP	Protocolo de Transferencia de Archivos (File Transfer Protocol o FTP).
GDA	Agente de directorio global (Global Directory Agents)
GUI	Interface de usuario gráfico
HDLC	High-Level Data Link Control
ICCCM	Manual de convenciones y comunicaciones del intercliente (Interclient Communications Conventions Manual)

IDL	Lenguaje de Definición de Interface (Interface Definition Language)
IEEE	Instituto de Electricidad e Ingeniería Eléctrica (The Institute of Electrical and Electronic Engineers)
IPC	Comunicación interprocesos
IPX	Internet Packet Exchange
ISDN	Integrated Services Digital Network
ISO	Organización Internacional para la Estandarización (International Organization for Standardization)
LAN	Redes de área local (local area network)
LU6.2	Unidad lógica de IBM 6.2
MAC	Control de acceso a los medios (Media Access Control)
MAN	Redes de área metropolitana (Metropolitan Area Network)
MAP	Protocolo de Automatización de Manufactura
MIPS	Millones de instrucciones por segundo
MLFLO	Millones de operaciones de punto flotante por segundo
PS	
NASA	The National Aeronautics and Space Administration
NDIS	Especificaciones de interface del controlador de la red de trabajo (Network Driver Interface Specification)
NetBEUI	Protocolos de Interface de usuario extendido NetBIOS (NetBIOS Extended User Interface)
NetBIOS	Sistema de entrada/salida básico de red de trabajo (Network Basic Input Output System)
NFS	Sistema de archivo de red (Network File System).
NFS	The National Science Foundation
NIC	Tarjeta de interface de red (network interface card)
NITS	Instituto Nacional de Ciencia y Tecnología (National Institute of Science and Technology)
NOS	Sistema operativo de la red (network operating system)
NPC	Netware core protocol
ODI	Interface de enlace de datos abierto de Novell
ODINS	Especificaciones de soporte de red de interface de enlace de datos abierto (Open Data-Link Interface Network support specification)
UP	
OMG	Agrupación para la administración de Objetos (Object Management Group)
ONC	Computación de redes abiertas (Open Network Computing)
OSF	Fundación de software abierto (Open Software Foundation)
OSI	Modelo de referencia (Open Systems Interconnection)
PC	Computadora personal (personal computer)
PE	Elementos de procesamiento (processing elements)
PEPS	Primero en entrar primero en salir
POSIX	Interface de sistemas operativos portables (Portable Operating System Interface)
RFT	Requerimientos para Tecnología (Requests For Technology)
RIP	Routing Information Protocol
RISC	Conjunto de instrucciones reducidas de computadora (Reduce Instruction Set Computer)
RND	Directorio de Nombre Remoto
SAA	Arquitectura de aplicación de sistemas (systems application architecture)

SAP Service Advertising Protocol
 SDLC Synchronous Data Link de SNA
 SDLC Control de enlaces de datos sincronos de alto desempeño de IBM
 SMB Bloques de mensajes de servidores (Server Message Blocks)
 SMP Multiprocesamiento simetrico (Symetric MultiProcessing)
 SMP Multiprocesamiento simétrico (Symmetric Multiprocessor)
 SMTP Protocolo de Transferencia de Correo Simple (Simple Mail Transfer Protocol)
 SNA Arquitectura de red de trabajo de sistemas IBM (Systems Network Architecture)
 SPX Intercambio de paquetes en secuencia (Sequenced Packet Exchange)
 SQL Lenguaje de Consulta Estructurado (Structured Query Language)
 SSI Imagen de sistema único (Single System Image)
 TCOC El Comité de Sistemas Abiertos (Technical Committee on Open System)
 TCP/IP Transmission Protocol/Internet Protocol
 TDM Multiplexación por división del tiempo (TDM)
 TDMA Acceso múltiple de división del tiempo
 TOP Protocolo de oficina y Técnico
 TPS Número de transacciones por segundo
 UI Unix International
 WAN Redes de área amplia. (Wide Area Network)
 WAN Redes de área extendida (Wide Area Networks)
 WORM una sola escritura y varias lecturas (Write-Once-Read-Many)

INTRODUCCIÓN

En los últimos años, se ha visto la transformación de las computadoras personales en estaciones de trabajo programables dentro de redes de área local e integradas con mainframes. Esta transformación se ha combinado con los avances tecnológicos en las minicomputadoras, interfaces gráficos, redes y comunicaciones. Los usuarios tienen la capacidad de interconectarse con varias plataformas eficientemente y transparentemente para distribuir la información y aplicaciones a través de sistemas heterogéneos y redes. Los usuarios están decididos a reducir hardware y desarrollo de software y costos operacionales trasladando sus aplicaciones desarrolladas en mainframes haciéndolas más eficientes, menos costosas, y transformándolas en poderosas estaciones de trabajo. Este nuevo modelo de computación es modelo computacional cliente/servidor y basado en arquitectura de cooperación y procesos distribuidos. Una gran cantidad de organizaciones está considerando moverse hacia ambientes distribuidos, procesamiento cooperativo, sistemas abiertos y arquitectura cliente/servidor.

Hoy en día el mundo de las redes de computación es una de las áreas más importantes en la informática; es una área que tiende a expandirse a nivel mundial y en nuestro medio. Durante la última década, el funcionamiento de las redes de computadoras ha cambiado enormemente; se han acordado una serie de normas internacionales para describir las arquitecturas de redes; en un futuro próximo, la capacidad de comunicarse con cualquier otro fabricante se podrá realizar sin mayor esfuerzo, estimulando así un mayor uso de las redes de computadoras.

Este estudio tiende a llenar un vacío o a reforzar los conocimientos adquiridos, ya que a medida que avanzamos hacia los últimos años de este siglo, se ha dado una rápida convergencia de esta área, organizaciones con oficinas dispersas en una amplia área geográfica esperan tener información en forma habitual, simplemente oprimiendo una tecla.

CAPITULO 1

INTRODUCCIÓN A LOS SISTEMAS DISTRIBUIDOS Y REDES DE COMPUTACIÓN.

1.1 EVOLUCIÓN HACIA LA ARQUITECTURA CLIENTE SERVIDOR

Es tan familiar la frase "Cliente/servidor es el camino de los 90's" ? Se ha dicho que la tecnología cliente/servidor eliminará las aplicaciones backlogs¹, reduciendo los costos de mantenimiento de software, incrementando su portabilidad, mejorando el rendimiento de los sistemas y las redes, y hasta eliminando la necesidad de minicomputadoras y mainframes. En lugar de profundizar en un debate acerca de cuáles de estas promesas son realistas, si es que alguna de ellas lo es, será mejor revisar la evolución de los ambientes de computación y definir un lugar para el modelo cliente/servidor; éste es el propósito de esta tesis. El modelo cliente/servidor cubre aspectos sobre diseño de aplicaciones, bases de datos, redes de computación y sistemas operativos.

1.1.1 PROCESAMIENTO BASADO EN COMPUTADOR CENTRAL (HOST)

El procesamiento de las aplicaciones basadas en host (computador central) es ejecutado en un solo sistema de computación, en el cual tiene terminales "tontas" conectadas.

Un ejemplo puede ser una computadora personal (PC) sola, o un mainframe que tiene conectadas terminales y su despliegue está basado en ambiente caracter. El procesamiento basado en host es totalmente no distribuido.

En los sistemas basados en un Host, la información y los componentes de la aplicación residen y son ejecutados sobre una plataforma de computación centralizada, que es utilizada por un número simultáneo de usuarios, (ver figura No 1.1).

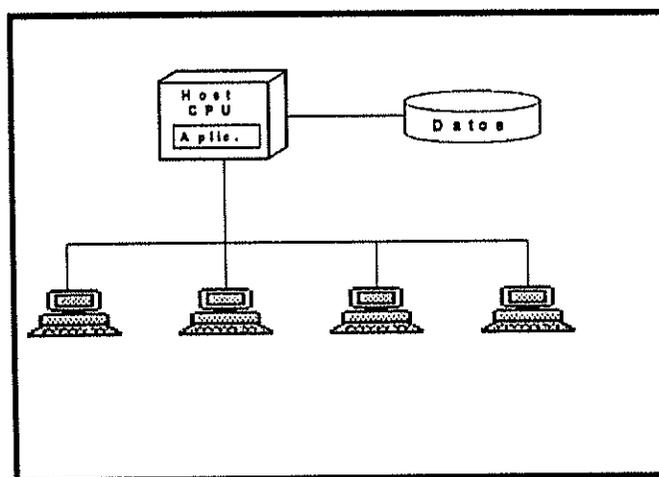


Fig. No. 1.1 Ambiente de procesamiento basado en un Host.

¹ acceso remoto a un computador central, emulando una terminal de un sistema centralizado.

1.1.2 PROCESAMIENTO MAESTRO ESCLAVO

El siguiente nivel del procesamiento de las aplicaciones distribuidas es el procesamiento maestro esclavo, como su nombre lo indica; en los sistemas maestro-esclavo, las computadoras son conectadas a una computadora maestra y ejecutan funciones de aplicación y procesamiento relacionados, sólo como lo ordena el computador central.

La aplicación de procesos en ambientes de sistemas maestro-esclavo es de alguna manera distribuido, aun cuando la distribución del proceso tiende a ser unidireccional, desde la computadora maestra hacia las computadoras esclavas. Típicamente, las computadoras esclavas tienen la capacidad de procesar aplicaciones con un procesamiento local limitado, tales como: la validación de un campo en la pantalla, edición y procesamiento de funciones de una tecla, (ver figura 1.2). Un ejemplo de procesamiento en un ambiente maestro esclavo es una computadora mainframe (host) que usa controladores de cluster (grupo), y terminales inteligentes.

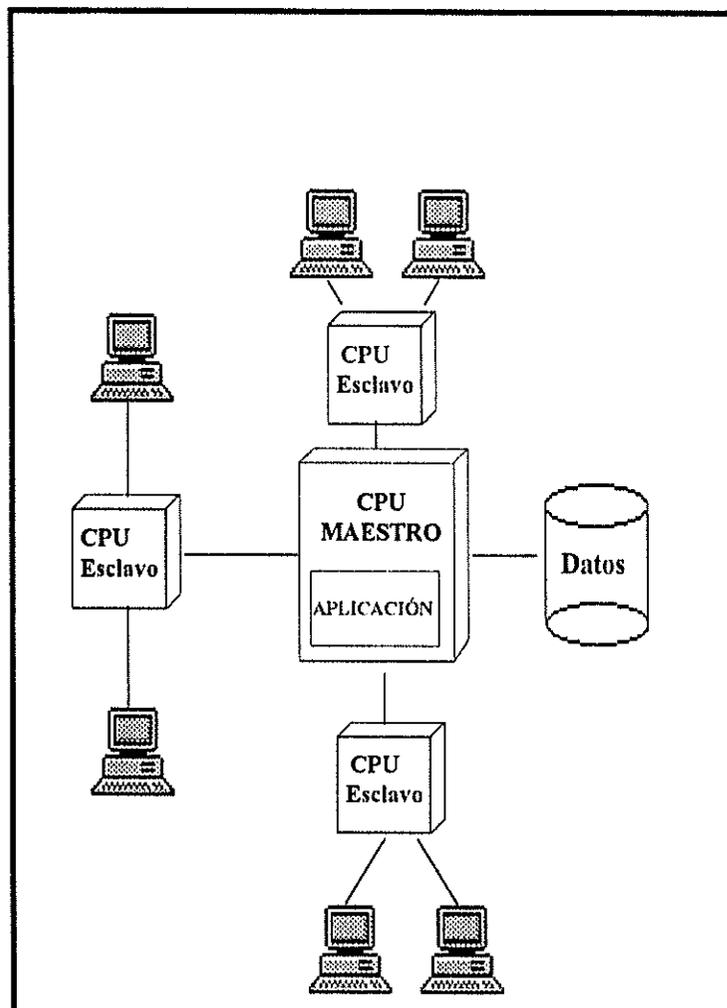


Fig. No. 1.2 Ambiente de procesamiento Maestro-Eslavo

1.1.3 PROCESAMIENTO CLIENTE/SERVIDOR

El modelo de computación cliente/servidor implica procesamiento cooperativo de requerimientos solicitados por un cliente o un solicitante, al servidor que procesa estos requerimientos y retorna el resultado a los clientes, (ver figura No. 1.3).

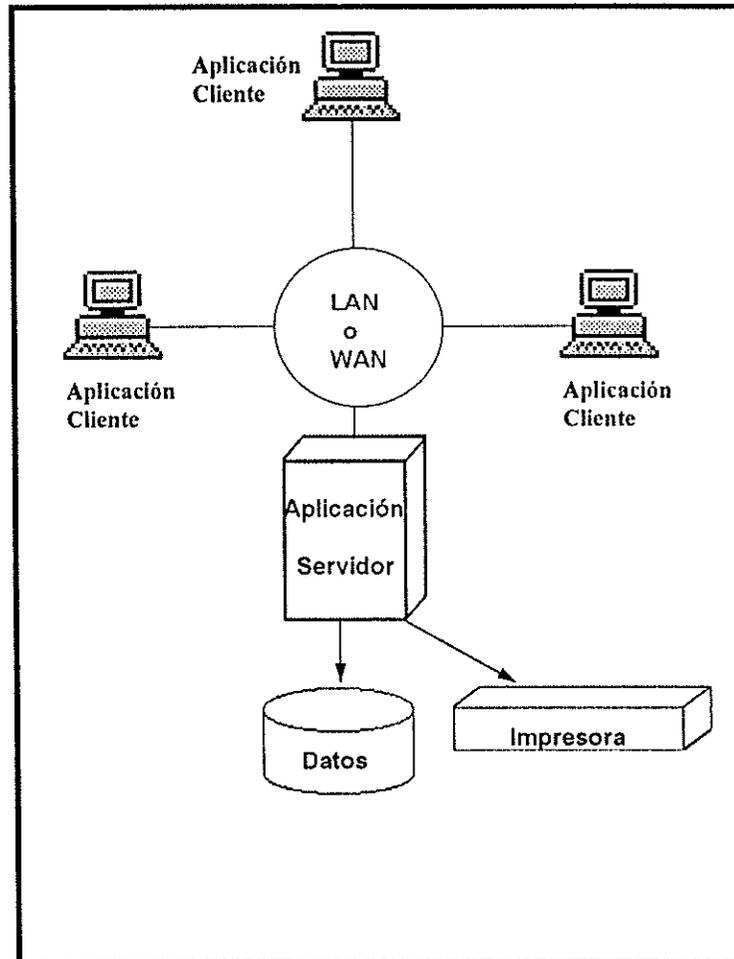


Fig. No. 1.3 Ambiente de procesamiento Cliente/Servidor

El proceso cooperativo cliente/servidor es realmente una forma especial de procesamiento distribuido, en donde los recursos (y tareas afectando los recursos), se encuentran distribuidos a través de dos o más modelos de computación discretos. Mientras que los sistemas distribuidos son relativamente un nuevo fenómeno, el nivel de distribución de los sistemas operativos es bien conocido y ampliamente usado.

El modelo cliente/servidor ha emergido de un alto nivel de procesamiento de dispositivos compartidos (ver figura No. 1.4), encontrados típicamente en redes de área local (LAN). En ambientes de procesamiento de dispositivos en redes de área local, donde computadoras personales (PCs) son conectadas a un dispositivo del sistema que permite a esas PCs, compartir recursos comunes, un archivo en un disco duro, una impresora, en la terminología LAN, tales dispositivos compartidos son llamados servidores, (un servidor de archivos, un servidor de impresora). El nombre servidor es apropiado, dado que esos

dispositivos compartidos son usados para recibir requerimientos y proporcionar servicios a las PCs, en funciones de bajo nivel en general.

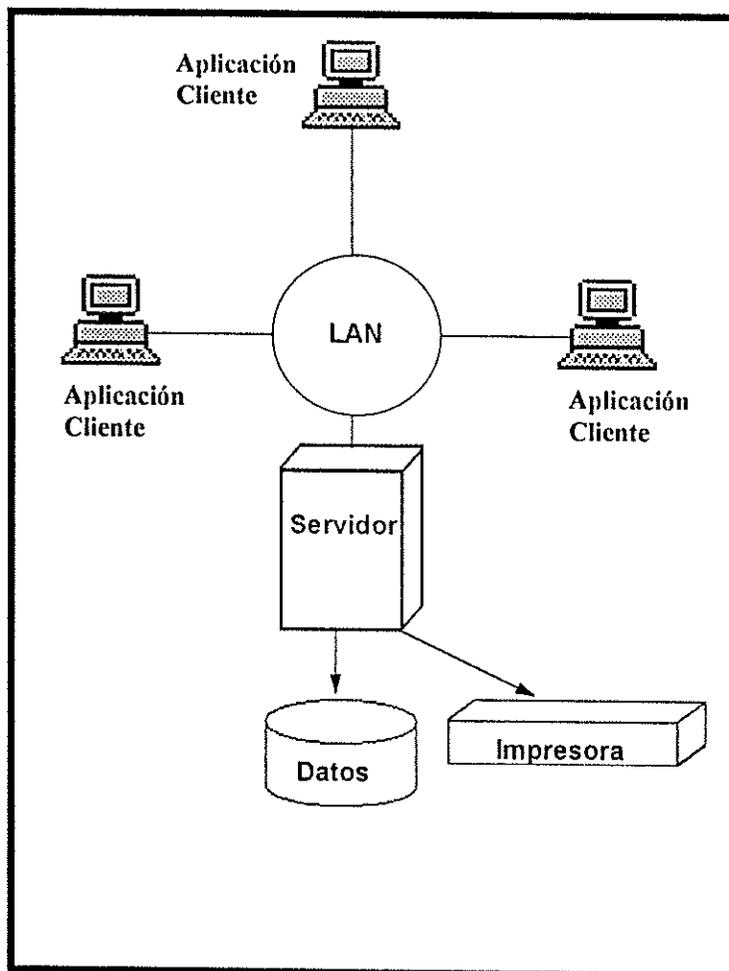


Fig. No. 1.4 Ambiente de procesamiento de dispositivos compartidos

En una típica red LAN, basada en procesamiento de dispositivos compartidos, los requerimientos de las PCs son generalmente limitados a servicios relacionados a compartir archivos o procesos de impresión (un archivo puede ser leído por varias PCs, y algunas páginas pueden ser mandadas de varias PCs a la misma impresora). Es obvio que la desventaja de tal aproximación es que el procesamiento de toda la aplicación es ejecutado de manera individual por cada PC y solamente ciertas funciones (impresiones, E/S de archivos) son distribuidas. Por lo tanto, un archivo completo tiene que ser enviado a la PC que hizo el requerimiento de lectura de este archivo. Si un archivo tiene que ser actualizado, el archivo completo es bloqueado por la PC que emitió el requerimiento de actualización.

Ejemplos de procesamiento de dispositivos compartidos que permiten a las redes de área local a tener un sistema dedicado exclusivamente para servicios de archivos e impresión son: NetWare de Novell y Manejadores LAN de Microsoft.

El modelo de procesamiento cliente/servidor es una extensión natural del procesamiento de dispositivos compartidos. Como las redes de área local crecieron en tamaño y número de estaciones

soportadas, el evolucionario sistema de dispositivo compartido; es éste un servidor de archivos o impresión; también creció en capacidad y poder. Gradualmente esos servidores llegaron a tener la capacidad de prestar servicio a un gran número de estaciones de trabajo. Al mismo tiempo, el papel de las estaciones de trabajo fue cambiando, de estaciones de trabajo y llegaron a convertirse en cliente de los servidores. La razón principal para el cambio fue que en un gran ambiente LAN, el compartir servicios de archivos e impresión, entre las estaciones de trabajo en un grupo de LAN, representaban solamente una fracción de una aplicación típica. La parte significativa de la funcionalidad de la aplicación fue también un buen candidato a compartir entre usuarios de LAN; así algunas de las aplicaciones de procesamiento fueron distribuidas a nuevos servidores, así como el servidor que recibe los requerimientos desde aplicaciones que están corriendo en estaciones de trabajo (clientes) y teniendo procesos para cada uno de los clientes.

En este modelo, el procesamiento de la aplicación es dividida entre el cliente y el servidor. El procesamiento es iniciado y controlado parcialmente por el requerimiento de servicio del cliente, y no como un maestro-esclavo; en lugar de esto, el cliente y el servidor cooperan para ejecutar exitosamente la aplicación; un servidor de base de datos, como Sybase o Microsoft SQL Server, son ejemplos de un ambiente de procesamiento cliente/servidor.

Una de las ventajas de la tecnología cliente/servidor puede ilustrarse por medio de la comparación de un servidor de archivos y un servidor de base de datos. Por ejemplo, si una aplicación de la PC necesita registros particulares de un archivo compartido, ésta envía un requerimiento para leer el archivo entero hacia el servidor de archivos, el cual hace que el archivo esté disponible para la PC. La aplicación de la PC tiene que buscar los registros seleccionados en el archivo, los recursos del servidor de archivos son usados para procesar el archivo completo, mientras que los recursos de la PC son usados para correr la aplicación que lee cada registro del archivo y seleccionarlos; si cada registro del archivo es enviado a la PC para ser procesado, una significativa porción de los recursos disponibles es utilizada ineficientemente, y las líneas de comunicación son sobrecargadas. En el caso del servidor de base de datos, una aplicación, que corre en la PC, envía el requerimiento de lectura de los registros al servidor de la base de datos. El servidor de la base de datos procesa el archivo, localizando y enviando solamente los registros requeridos por la aplicación de la PC. Ambos (el cliente y el servidor) computan cooperativamente los recursos para ejecutar la consulta (query) requerida.

El procesamiento cliente/servidor requiere :

- confiabilidad, comunicación robusta entre clientes y servidores,
- interacción cooperativa cliente/servidor que es iniciada por el cliente,
- distribución del procesamiento de aplicaciones en los clientes y servidores,
- control reforzado del servidor sobre qué servicios o información pueden requerir los clientes,
- el servidor debe ser un arbitro para los requerimientos conflictivos de los clientes.

1.1.4 PROCESAMIENTO PUNTO A PUNTO

El modelo cliente/servidor distingue entre los clientes que requieren servicios y los servidores que prestan servicio a esos requerimientos. Todos los participantes de los sistemas de procesamiento punto a punto, como quiera que sea, son iguales y pueden requerir y proveer servicios hacia y desde cada uno, (ver figura No. 1.5).

El procesamiento es realizado en cualquier lugar donde se encuentren disponibles los recursos de computación, incluyendo dispositivos compartidos, unidad central de proceso (CPU) y memoria. Un simple sistema de procesamiento de punto a punto puede ser un cliente de otros servidores y servidor para otros clientes (incluyéndose él mismo), además, en el procesamiento inteligente punto a punto, un servidor puede distribuir la carga entre los servidores disponibles, y pueden aun optimizar tal distribución basada en la característica de la red y los servidores.

Idealmente, los ambientes punto a punto deben de proveer un procesamiento cooperativo transparente entre las aplicaciones residentes en una amplia variedad de plataformas de hardware y software. Una de las metas de los ambientes de procesamiento punto a punto es soportar redes de bases de datos, donde los usuarios podrán llegar a moverse libremente entre múltiples bases de datos heterogéneas. De este modo, la implementación práctica de los ambientes de computación punto a punto, es la meta de las arquitecturas y diseñadores de los sistemas distribuidos.

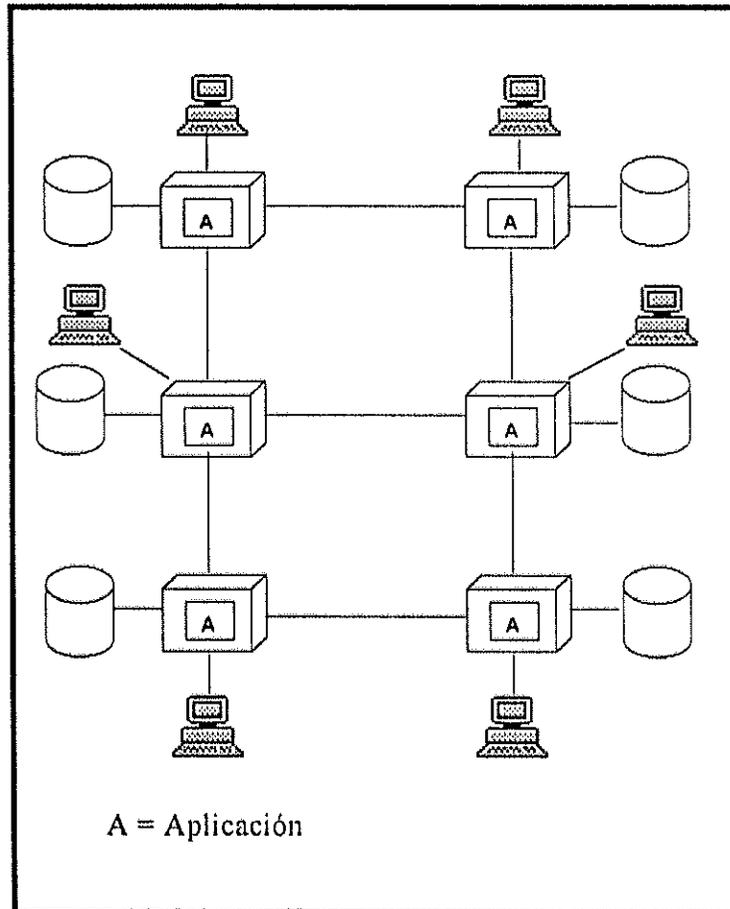


Fig No. 1.5 Ambiente de procesamiento punto a punto

1.1.5 VENTAJAS DE LA COMPUTACIÓN CLIENTE/SERVIDOR

Actualmente muy pocos productos soportan el procesamiento punto a punto. Hay un espectro de uso de técnicas salientes que hacen que la implementación del procesamiento punto a punto sea una tarea más que difícil. Mientras no exista el ideal punto a punto, es una aproximación razonable para el desarrollo de sistemas distribuidos. Este modelo provee beneficios significativos y puede ser utilizado en las soluciones de negocios de hoy.

Aparte de las estrictas consideraciones de mercadeo, más desarrolladores de la tecnología de habilitación de aplicación, están empezando a concentrarse en la tecnología cliente/servidor (incluyendo herramientas CASE²). Hay beneficios reales al adoptar la arquitectura cliente/servidor:

- Ésta permite a las corporaciones mejorar el nivel de la tecnología computacional de escritorio (desktop). Las estaciones de trabajo tienen considerablemente más poder computacional; anteriormente se disponía solamente de mainframes (computador central), y son éstas una fracción del costo de los mainframes.
- Permite que el procesamiento resida cerca de la fuente de los datos, que están siendo procesados. (La arquitectura cliente/servidor es una manera especial de procesamiento distribuido - procesamiento cooperativo). Por lo tanto, el tráfico de la red y el tiempo de respuesta puede ser reducidos en gran manera, incrementando la capacidad de transportar sobre la red una carga pesada. Por lo tanto, los requerimientos del ancho de banda de las redes y su consecuente costo pueden ser reducidos.
- Esta facilita el uso de interfaces de usuario gráfico (GUI) disponibles sobre las poderosas estaciones de trabajo. Estas nuevas interfaces pueden ser entregadas al cliente en una variedad de técnicas de presentación visual junto con una fácil navegación y una consistencia estándar flexible. Como resultado, la inversión en entrenamiento y educación pueden ser mejorados; nuevos productos que exceden las expectativas del cliente, pueden ser desarrollados más rápidamente y la resistencia del usuario final de aceptar nuevos productos puede minimizarse.
- Esta permite estimular la aceptación de sistemas abiertos. El hecho de que los clientes y servidores puedan llegar a ejecutarse en diferentes plataformas de hardware y software, y permitan a los usuarios liberarse de una arquitectura propietaria, tomando entonces ventaja económica, de mercado, competitiva del mercado abierto de productos disponibles.

El modelo cliente/servidor no es perfecto; hay más de una desventaja usando este modelo, por ejemplo :

- Si una porción significativa de la lógica de la aplicación es movida hacia el servidor, el servidor puede llegar a convertirse en un cuello de botella en la misma forma que un mainframe en la arquitectura maestro-esclavo. Los recursos limitados de los servidores llegarán a su máxima demanda debido al creciente número de consumidores de recursos (usuarios finales).
- Las aplicaciones distribuidas, especialmente aquellas que son diseñadas para procesamiento cooperativo, son más complejas que las no distribuidas; esto es cierto para el desarrollo de las aplicaciones en ambientes de corrida real, y las herramientas usadas para el manejo de estos ambientes distribuidos. Sin

² Ingeniería de Software ayudada por computador (CASE o Computer Aided Software Engineering)

embargo, algo de esta complejidad puede ser eliminada reduciendo un problema grande y convertirlo en un conjunto de problemas más pequeños, posiblemente problemas interdependientes, similares al diseño de sistema modular.

Hay afirmaciones hechas por algunos vendedores y expertos relativos a la industria cliente/servidor; algunas de ellas, son probablemente ciertas, mientras que otras se asemejan más a estrategias de ventas y mercadeo; por ejemplo, el que sostengan que el cliente/servidor convierta a los usuarios no técnicos en profesionales en el desarrollo de software, es altamente cuestionable. Lo mismo ocurre con lo que se dice acerca de que el cliente/servidor forzará a las minicomputadoras y mainframes a desaparecer.

Sin embargo, es muy probable que la implementación del modelo cliente/servidor reducirá el costo de mantenimiento del software, incrementará la portabilidad del software y rendimiento de las redes existentes, y aun eliminará las aplicaciones acceso remoto (backlog), por el incremento de la productividad de los desarrolladores y reducirán el ciclo de vida del desarrollo.

Las publicaciones actuales de mercadeo enfatizan solamente de un lado del modelo cliente/servidor, el lado relacionado con la distribución de la información. La verdadera arquitectura cliente/servidor abarca mucho más que datos; esta incluye redes y comunicación de datos, presentación distribuida, procesamiento transaccional distribuido, estándares y sistemas abiertos.

1.2 ESTÁNDARES Y SISTEMAS ABIERTOS

Antes de discutir todos los aspectos de la arquitectura cliente/servidor, éstos serán útiles para ver la diferencia entre una arquitectura y un producto.

Una arquitectura es un conjunto de definiciones, reglas y términos que son usados como guías para construir los productos. Un producto es una implementación específica de una arquitectura. La arquitectura sola, sin la implementación de un producto, no es muy útil. Un producto que implementa la arquitectura hace usable la arquitectura.

Una arquitectura bien diseñada debería de ser implementada en uno o más productos, y debería de tener un buen tiempo de vida, a través de un producto específico. Para alcanzar esas metas, una arquitectura debe estar basada sobre estándares de la industria y consecuentemente, se puede inicializar la introducción de nuevos estándares. Productos que implementan tales arquitecturas deben promover y soportar la estandarización. El modelo de computación cliente/servidor es una arquitectura, que todavía no ha madurado. Los productos que implementan la arquitectura cliente/servidor, aunque ofrecen características muy útiles, difieren significativamente en funcionalidad, interfaces y aun en el nivel de flexibilidad a los estándares que deberían regular la arquitectura. No todos los estándares relevantes para la arquitectura cliente/servidor están finalizados o completamente desarrollados a la fecha. Por lo tanto, la discusión de la arquitectura cliente/servidor empieza con ver hacia los estándares relevantes de la industria, especialmente los relacionados con los sistemas abiertos.

1.2.1 SISTEMAS ABIERTOS

Desde el punto de vista de la tecnología de la información, una empresa típica de negocios es una colección de diferentes sistemas, que residen sobre una amplia variedad de plataformas de hardware y software de múltiples vendedores, mutuamente incompatibles, que se corren en miles de negocios, y aplicaciones científicas y de ingeniería. Hoy los clientes desean unir sus empresas en un sistema completo coherente, tratando de unir todos sus sistemas. Independientemente de la plataforma del vendedor en la cual esos sistemas están corriendo. Más usuarios quieren extender sus negocios uniendo sus empresas en sistemas amplios con sus proveedores, distribuidores y sus clientes. Para tener esta interconexión de todos los sistemas y mover las aplicaciones de un sistema a otro, esos sistemas propietarios existentes deben de abrir sus arquitecturas adoptando interfaces basados en estándares. Aun los gigantes de los sistemas propietarios como IBM, reconocen el deseo de interoperabilidad de los clientes, a través de múltiples plataformas de diversos vendedores.

Interoperabilidad significa la existencia múltiples vendedores de sistemas, los cuales deberfan poder hablar (comunicarse) uno con otro. Interoperabilidad abierta significa que nuevas aplicaciones deberfan construirse en este camino, de tal manera que ni el diseño, ni la implementación del sistema deberfa de cerrar la aplicación a un vendedor particular de plataforma de hardware o software. En otras palabras, las aplicaciones de sistemas abiertos deben de ser portables a través de todas la plataformas especificadas por los clientes, para evitar encerrarse en un sistema en particular. Como las aplicaciones de negocios crecen en alcance, los clientes deberfan ser capaces de moverse de estas aplicaciones, hacia un sistema más grande y poderoso, sin la necesidad de modificar sus aplicaciones; un sistema abierto deberfa de facilitar la escala de aplicaciones para el desempeño y número de operaciones por unidad de tiempo (throughput)

Por supuesto, en la medida en que los sistemas abiertos llegen a una completa interoperabilidad y cumplan con los requerimientos de portabilidad de aplicaciones, éstos deben construirse de acuerdo con los estándares aceptados por la industria. Solamente cuando un sistema conforma los estándares de la industria, éste puede interoperar con otro sistema flexible estándar, y solamente las aplicaciones son portables de un sistema a otro.

Es importante notar que los sistemas abiertos no significan automáticamente el sistema operativo UNIX. La apertura no es la función del sistema operativo. Los sistemas abiertos deberfan tener las siguientes características:

- Flexible con los estándares de la industria, para programación, comunicación, redes, administración del sistema, presentación, servicios del sistema e interfaces entre las aplicaciones y los servicios del sistema.
- Portabilidad de las aplicaciones a través de los sistemas.
- Escalabilidad del rendimiento de las aplicaciones y el número de aplicaciones por unidad de tiempo.
- Interoperabilidad a través de los sistemas.

En efecto, El Instituto de Electricidad e Ingeniería Eléctrica (The Institute of Electrical and Electronic Engineers o IEEE), El Comité de Sistemas Abiertos (Technical Committee on Open System o TCOC) tienen la siguiente definición de sistemas abiertos: "Un conjunto comprensible y consistente de estándares de tecnología de información internacional y un perfil estándar funcional que especifique

interfaces, servicios y que soporte formatos que acompañen la interoperabilidad y portabilidad de las aplicaciones, información y personas”³.

1.2.2 BONDADES Y ESTÁNDARES PROPIETARIOS

Alguno de los clientes pueden estar interesados solamente en la portabilidad, escalabilidad e interoperatividad de la aplicación. Esta posición es usualmente justificada, aduciendo que estas características, traen un alto retorno y beneficio de la inversión. Sin embargo, sin el acompañamiento de los estándares, el gran potencial de los sistemas abiertos no se puede alcanzar. Por ejemplo : una de las metas formuladas de la arquitectura de aplicación de sistemas (SAA) de IBM, es proveer portabilidad, consistencia y conectividad de la aplicación a través de todas las SAA soportadas por las plataformas MVS, VM, AS/400 y OS/2. Sin embargo, mientras que los beneficios son claros, SAA no provee interoperabilidad entre las plataformas IBM y no IBM, aunque así se limiten las bondades de los condescendientes sistemas SAA.

Sin embargo, todos los sistemas SAA fueron diseñados para cumplir con los estándares relevantes de la industria, por lo tanto, los sistemas SAA han sido realmente abiertos a todos los vendedores y usuarios adherentes a esos estándares. Los estándares en cuestión son aquellos para programación, comunicación, redes, administración de sistemas, presentación, servicios del sistema, e interface entre aplicaciones y los servicios del sistema.

Hay muchas ramificaciones importantes de amplia proliferación de estándares propietarios :

- Los clientes que están encerrados con sistemas de un solo proveedor y están planificando cambiarse de proveedor/sistema/arquitectura, están afrontando altos costos de cambio de un vendedor a otro.
- Como una regla, los usuarios de sistemas propietarios han escrito una gran cantidad de software y documentación y han entrenado a sus empleados teniendo destreza en un sistema en específico. No todas esas inversiones pueden ser salvadas cuando se migra de un ambiente a otro.
- Proveedores Independientes de software tienden a desarrollar software para vendedores con grandes bases de datos instaladas, dándole a los vendedores de sistemas grandes, ventajas competitivas adicionales.

Muchos usuarios están forzados a mantenerse en los sistemas propietarios. El problema real surge cuando los proveedores de sistemas propietarios no conocen a fondo las necesidades de los clientes. Las necesidades de los clientes del siglo veintiuno cambiará y crecerán dramáticamente.

En los últimos años, las redes mundiales a gran escala, que tratan de conectar cientos de procesadores host, estaciones de trabajo, servidores y PCs han empezado a ser desarrollados. Dado que las redes típicas están compuestas por una amplia gama de proveedores de sistemas, la estructura de tal magnitud, redes heterogéneas prueban ser extremadamente complejas y el manejo de la red llega a ser muy difícil. Estos hechos resultan en una necesidad mundial para la aceptación de sistemas abiertos basados en estándares. La arquitectura cliente/servidor y su implementación no son la excepción: los estándares permitirán a los clientes y servidores de diferentes proveedores comunicarse, portar aplicaciones de una plataforma a otra, sin abandonar la arquitectura cliente/servidor, mover las aplicaciones a sistemas de diferentes tamaños y alcanzar rendimiento a escala.

³ Fuente : Client/Server Architecture, Alex Berson, McGraw-Hill, Pag. 14

Entre las organizaciones de estándares que juegan papeles claves en los sistemas abiertos son :

- POSIX : Interface de Sistemas Operativos Portables (Portable Operating System Interface) para ambientes de computación, han sido definidos por IEEE TCOS para proveer gran consistencia a través de diferentes sistemas operativos.
- X/Open es una organización no lucrativa fundada en 1984 para resolver problemas causados por software e incompatibilidad de sistemas. X/Open juega un papel importante en la definición de Ambientes de Aplicación Común (CAE, Common Aplicacion Enviroment).
- ISO, La Organización Internacional para la Estandarización (International Organization for Standardization) tiene como una meta el desarrollo, aceleración y promoción varios estándares y productos que ellos implementan. Entre ellos está el modelo de referencia OSI (Open Systems Interconnection).
- COS, Corporación para Sistemas Abiertos (Corporation for Open Systems) concentra sus esfuerzos en el área de OSI, ISDN (standars conformance testing and certification).
- OMG, Agrupación para la Administración de Objetos (Object Management Group) es una organización internacional envuelta en el desarrollo de software y sistemas de tecnología orientada a objetos.
- OSF, la Fundación de Software Abierto (Open Software Foundation) y UI (Unix International) son de los dos mejores proveedores de tecnología para ambientes de sistemas abiertos.

1.2.3 BENEFICIARIOS DE LOS SISTEMAS ABIERTOS

La meta final de un ambiente computacional abierto ideal es hacer posible la distribución real de redes de trabajo en donde :

- El cómputo puede ocurrir transparentemente a través de toda la red.
- Aplicaciones, recursos, funcionalidad y el poder de ser compartidos a través del ambiente.
- Los usuarios estarán provistos con amplias posibilidades de portabilidad, interoperabilidad y la escalabilidad de aplicaciones.

La creación de tales ambientes de computación abiertos no significa la eliminación de sistemas operativos propietarios, ni tampoco significa que los sistemas abiertos deben ser puramente UNIX. Los ambientes de sistemas abiertos deben ir acompañados de interoperabilidad y portabilidad de aplicaciones y de datos, y dar apoyo a un conjunto completo y comprensivo de estándares de funcionalidad y tecnología aceptado internacionalmente. Realmente, los sistemas abiertos pueden coexistir con los sistemas operativos propietarios. Esta coexistencia debe ir acompañada con puntos de interfaces y conexión, así los vendedores de sistemas cerrados son evitados y las investigaciones en tecnología existente, aplicaciones, entrenamiento son mejorados. Realmente, los ambientes de computación abiertos benefician a los proveedores de sistemas, a los proveedores de software independiente y a los usuarios finales de las siguientes manera:

- Los vendedores de sistemas de hardware y software son beneficiados por los sistemas abiertos, debido a la reducción de costos del desarrollo de sistemas operativos, al acortar el tiempo de mercadeo de los nuevos productos. En lugar de gastar tiempo y esfuerzo en la creación de sistemas propietarios, los proveedores deben de ser capaces de mejorar la arquitectura, características y funcionalidad de nuevos productos.
- Los sistemas abiertos no eliminan la competencia. Recíprocamente, los proveedores compiten y agregan nuevos valores a sus productos, y logran la especialización de productos y de plataformas de hardware, calidad, servicio y la optimización de soluciones propietarias.
- Los proveedores de software independientes son beneficiados de los ambientes de computación abiertos, por el desarrollo de productos basados en estándares internacionales. La flexibilidad con interfaces estándares prácticamente garantizarán mercados más amplios para sus productos.
- Los usuarios finales serán los mayores beneficiarios de los ambientes de computación abiertos. Sus compras estratégicas de hardware permitirán comprar sistemas sin temor de efectuar compras erróneas. Los usuarios finales de los sistemas abiertos estarán asegurados de tener el mismo ambiente de operación con cualquier sistema que ellos compren, mainframe, minicomputadoras, estaciones de trabajo o supercomputadoras. La mezcla de proveedores de interoperabilidad y conectividad de redes, será simplificada. Los costos de desarrollo de software, entrenamiento y mantenimiento serán reducidos, y más productos estarán disponibles a precios más competitivos. Corporaciones enteras y unidades individuales de negocio tendrán mayor flexibilidad en escoger sus productos y sistemas que se acoplen mejor a sus requerimientos.

Mientras que los ambientes de computación verdaderamente abiertos, sean todavía la meta futura de muchos proveedores, el gremio estándar y los usuarios finales; al día de hoy se han hecho muchos avances alentadores que nos acercan al ambiente de sistemas abiertos, La Fundación de Software Abierto (OSF o Open Software Foundation) y UNIX Internacional, son dos organizaciones que han expresado, que el desarrollo de sistemas abiertos es su principal objetivo.

De hecho, hoy ambas organizaciones ofrecen versiones casi similares de ambientes de computación abiertos. OSF (Open Software Foundation) ofrece DCE (Distributed Computing Environment), mientras que la solución de UI (UNIX International) es ONC (Open Network Computing), en realidad ninguna organización, sin importar cuan grande sea o qué tan influyente sea, podría administrar la creación de un ambiente tan completo y complicado, y forzar a la industria a aceptarlo. Esta es la razón por la que tanto la OSF como la UI buscan la fortaleza entre sus asociados. Por eso es que los proveedores de software pertenecen a una o ambas organizaciones.

1.3 AMBIENTE DE COMPUTACIÓN DISTRIBUIDO (DCE o DISTRIBUTED COMPUTING ENVIRONMENT)

Aunque DCE (Distributed Computing Environment) y ONC (Open Network Computing) parezcan similares, DCE puede tener mayor aceptación. De hecho, la Comisión Europea (the coordinating body for Unified Europe) ha tomado a DCE (Distributed Computing Environment) de OSF (Open Software Foundation) sobre ONC (Open Network Computing) de UI (UNIX International), para el desarrollo de nuevas generaciones de aplicaciones distribuidas, como estrategia de computación de múltiples proveedores. La Comisión Europea mantiene que DCE es más completo que ONC, y que el soporte de la DCE es esperado para ONC.

Para propósitos prácticos, DCE y ONC son incompatibles solamente en un punto: en las llamadas a procedimiento remoto (RPC o Remote Procedure Calls). Esta incompatibilidad no es crítica; hay muchos proveedores que implementan DCE en los sistemas operativos estándar de UI: UNIX System V Release 4, frecuentemente abreviado SVR4.

1.3.1 ANTECEDENTES

El Ambiente de Computación Distribuido (DCE) de OSF (Open Software Foundation), le da poder a los usuarios y los desarrolladores de software, les dan ventajas en los sistemas realmente abiertos distribuidos. DCE provee el fundamento que es necesario para el uso y desarrollo de aplicaciones distribuidas en ambientes de hardware y software heterogéneo.

Para desarrollar DCE de la OSF, se utilizaron Requerimientos para Tecnología (RFT o Requests For Technology), a través de los procesos abiertos RFT, OSF ha seleccionado e integrado las mejores tecnologías disponibles al día. Mas de 50 proveedores respondieron al RFT. Las tecnologías seleccionadas se derivaron en su mayoría de la proposición DECorum, la cual fue suscrita conjuntamente por Hewlett Packard (HP)/Apollo, IBM, Digital Equipment Corporation (DEC), Microsoft Corporation, Locus y Transarc.

Al seleccionar la tecnología DECorum, OSF ha posicionado DCE como una opción. Para la Arquitectura de Computación de Redes de Sun Microsystems (NCA o Network Computing Architecture), NCA ha sido adoptado por Sun Microsystems en grandes bases de clientes (Sun Microsystems es miembro de UI).

Sin embargo, recientemente la IBM anuncio que en un futuro cercano, la interoperabilidad entre sus sistemas SAA (Systems Applications Architecture) y la versión del sistema operativo UNIX Advanced Interactive eXecutive (AIX), más la decisión de OSF para adoptar una parte significativa de AIX como una parte del Kernel del sistema operativo (OSF/1), hace una posibilidad muy real el desarrollo de puentes (bridges) entre SAA, AIX y DCE.

Tal y como los ambientes distribuidos abiertos penetran en los sistemas de mercado comercial, tradicionalmente dominados por IBM y productos compatibles con IBM, la interoperabilidad entre SAA y AIX conjuntamente con DCE, serán un importante requerimiento, así como uno de los factores decisivos para la aceptación de DCE.

1.3.2 ARQUITECTURA DCE

En términos generales, el ambiente de computación distribuido (DCE o Distributed Computing Environment) es un ambiente consistente integrado, que tiene los siguientes componentes :

- A) Sistema de archivo Distribuido (DFS)
- B) Servicio de directorio
- C) Llamadas de procedimiento remoto (RPC)
- D) Servicios de subprocessos
- E) Servicios de tiempo

A) SISTEMA DE ARCHIVO DISTRIBUIDO (DFS o DISTRIBUTED FILE SYSTEM)

Los sistemas de archivo distribuido permiten a los usuarios de un sistema conectado en la red, acceder y modificar datos almacenados en archivos que residen en otros sistemas. Desde el punto de vista de la arquitectura cliente/servidor, el cliente es el sistema donde el usuario está trabajando, mientras que el servidor es el sistema donde la información es almacenada.

Cuando los datos son accedidos por el servidor de archivos, una copia de los datos es almacenada en el sistema del cliente, y le permite leer y modificar esta información; la información modificada es devuelta al servidor de archivos para ser actualizada. Es obvio que el problema reside cuando múltiples usuarios tratan de acceder y modificar la misma información. Una solución para este problema es forzar al servidor de archivos a mantener un registro de los clientes y su información accesada. DCE DFS usa un conjunto de registros para mantener el control de la información accesada. Los registros son asignados al cliente por el servidor, cuando los clientes requieren la información; éste se basa por el tipo de acceso que el cliente requiere. Para modificar datos, el cliente debe requerir un acceso de escritura al servidor. Una vez que el registro de escritura es asignado, el servidor puede informar a los otros clientes, que el registro de escritura es usado. Si otros clientes tienen la misma información solamente para propósitos de lectura, ellos pueden ser notificados que su registro es revocado o liberado.

DCE DFS provee las siguientes características avanzadas de sistemas de archivos distribuidos :

- **SEGURIDAD DE ACCESO Y PROTECCIÓN** : mientras que en los sistemas originales UNIX fue notoria su débil seguridad, DCE DFS implementa seguridad reforzada dando soporte a la autenticidad de usuarios (el sistema Kerberos permite a los clientes intercambiar información encriptada con la autenticación del servidor, así los clientes pueden transportar etiquetas Kerberos, el cual verifica al cliente y si el mismo tiene derecho) y tiene un lista de mecanismos de control de acceso, para conceder acceso de archivos a clientes autorizados.
- **CONFIABILIDAD DE DATOS** : mientras que los sistemas distribuidos permiten teóricamente la eliminación de fallas de punto sencillo, los sistemas distribuidos diseñados incorrectamente, pueden forzar a un cliente a confiar en un número crítico de recursos; la pérdida de uno de estos tendría como resultado la incapacidad del cliente a seguir procesando. Para prevenir este problema, DCE DFS soporta la replicación para todos los servicios de redes: Si uno de los servidores llegara a no estar disponible, el cliente automáticamente cambia de un servidor a otro replicado.

- **DISPONIBILIDAD DE LOS DATOS** : DCE DFS permite, al administrador del sistema, ejecutar rutinas de mantenimiento (tales como copia de seguridad de la información, movimiento de archivos, etc.), en los recursos de la red, sin provocar que cualquiera de los servidores de la red se queden fuera del sistema.
- **DESEMPEÑO**. DCE DFS : es un sistema eficiente y extensible, por mantener la información del estado de los archivos y los datos en el sistema de un cliente. DFS está reduciendo el número de requerimientos de datos de los clientes, y así reducir los servidores y la carga en la red.
- **MANEJABLE**. DCE DFS : utiliza bases de datos distribuidas, para mantener una copia de la localización de archivos, autenticación, y la lista de control de accesos, que es usada por los clientes y servidores. Estas bases de datos son divididas y separa la administración y mantenimiento de dominios, las cuales pueden ser accedidas por cualquier cliente. Adicionalmente estas bases de datos son autoconfigurables y fáciles de operar.
- **ADAPTACIÓN ESTÁNDAR** : DCE DFS se adapta con el POXIS 1003.1 estándar de semántica del sistema de archivos de la IEEE.
- **INTEROPERABILIDAD CON NFS** (Network File System) : DCE DFS proporciona pasarelas (gateways) que permiten a los clientes que usan NFS interoperar con servidores DCE DFS, y proporcionar un camino de migración de NFS a DCE DFS.

B) SERVICIO DE DIRECTORIO

En las redes de computadoras, al igual que las personas, se requiere de nombres y direcciones, para describir, registrar y encontrar las características de varios servicios e información que ellos proporcionan. Como en los sistemas de correo real, un sistema de correo electrónico debe estar disponible, para localizar el buzón de los clientes para enviarles el correo.

En los ambientes de computación distribuido, las aplicaciones de envío de correo estarán en contacto con aplicaciones de servicio de nombres o directorios, para obtener el nombre del usuario y su dirección. En los DCE, cualquier aspecto que pueda ser llamada y accedida individualmente (es decir servicios de redes, correo electrónico, computadoras, etc.) es llamada un objeto. Cada objeto tiene su correspondiente entrada en el directorio de servicios. Cada entrada contiene atributos que describen el objeto. El nombre de cada entrada es agrupada en listas llamadas directorios. En los DCE, los directorios pueden estar organizados en jerarquías, en las cuales un directorio puede contener a otro directorio.

Los nombres y los servicios de directorio están centralizados en los DCE. La razón de esto es porque todos los objetos de DCE son definidos por su nombre, su aplicación y su servicio, para acceder a los objetos, se hace accedando la lista de entradas, y se obtienen sus atributos. Así, las características del objeto son separadas del mismo, y lo más importante, la independencia de localización de los objetos es asegurada. Así como las organizaciones permiten a las aplicaciones y servicios a acceder objetos, aun si los objetos son movidos o cambiados en varios de sus atributos.

Los servicios de directorio DCE están integrados con otros componentes DCE, incluyendo DCE DFS, y poseen las mismas características avanzadas (seguridad, confiabilidad, disponibilidad, manejabilidad, rendimiento) como los DCE DFS.

En un ambiente cliente/servidor DCE, un cliente local en alguna parte de la red DCE (en un dominio), necesita encontrar el nombre de un cliente remoto, le envía este requerimiento a un GDA (Global Directory Agents) local, que reside en un servidor de nombres. El servidor donde se encuentra el GDA (Global Directory Agents) manda el requerimiento al mundo por medio del servicio X.500, el cual busca el nombre y devuelve el resultado al GDA (Global Directory Agents), y este lo devuelve al cliente.

C) LLAMADAS DE PROCEDIMIENTO REMOTO (RPC)

La sintaxis, semántica y servicios de presentación de las llamadas a procedimiento remoto (RPC Remote Procedure Calls) representan una extensión de llamadas a subrutinas de un lenguaje de alto nivel. Las RPCs permiten que el código actual de un procedimiento llamado, resida y sea ejecutado físicamente en un procesador remoto y mantener transparente la aplicación. El mecanismo RPC es el aspecto más crítico en la arquitectura DCE (ambientes de computación distribuida). Éste actúa como un pegamento que mantiene unidos a todos los componentes DCE. La base de OSF DCE RPC es el sistema de computación de redes (NCS Network Computing System) de HP/apollo versión 2.0, de llamadas a procedimientos remotos, remitida como parte de la proposición DECorum. Las llamadas de procedimiento remoto de los ambientes de computación distribuido son fáciles de usar, y están diseñados para ser transparentes en varias arquitecturas de redes y soportar servicios de subprocesos descritos a continuación. La sintaxis, semántica y servicios de presentación es la mayor diferencia entre el DCE de OSF y ONC de UI. Un punto de mención importante es que la OSF está en desacuerdo con el camino de la ONC de alentar a los usuarios a modificar el protocolo fundamental RPC.

D) SERVICIOS DE SUBPROCESOS (THREADS)

Un ambiente de computación típico permite el logro de sus metas, y une todos los procesadores participantes. Por lo tanto, existen oportunidades para implementar un cierto grado de procesamiento paralelo. Hay una gran cantidad de estrategias para implementar código de procesamiento paralelo. OSF selecciona la estrategia de subprocesos (threads) para sus ambientes de computación distribuido; esta estrategia usa subprocesos existentes y operan dentro de una instancia sencilla de la ejecución de un programa y sus direcciones. El programa puede usar herramientas de sincronización especial, como semáforos, accesos de control común, modificar recurso compartidos por varios usuarios (por ejemplo, variables de memoria).

Hay muchos otros métodos para implementar el procesamiento paralelo (compartir memoria entre múltiples programas, o el uso explícito de verbos de sincronización para intercambiar mensajes entre varios programas). Sin embargo, esos métodos usualmente involucran recursos externos al programa. La arquitectura multitarea concertada (CMA o Concert Multithread Architecture) DCE, es la fundación de servicios de tareas de DCE. Esta ofrece portabilidad y soporta aplicaciones POSIX 1003 y especificaciones de interface de servicios de sistema.

E) SERVICIOS DE TIEMPO

La función de los componentes de servicio de tiempo es sincronizar los relojes de los nodos de todas las redes con el reloj que controla la red como uno solo. Debido a que es muy completo y simple, OSF seleccionó a los servicios de sincronización de tiempo distribuido de DCE.

EL RESULTADO DEL MODELO CLIENTE/SERVIDOR DCE

DCE de la OSF se acomoda en el paradigma cliente/servidor. Por lo tanto, los componentes DCE deben estar presentes en: el requeriente del servicio, el cliente DCE, el proveedor de servicio y el servidor DCE. (Ver la figura No. 1.6).

DCE no es un simple paquete de software que puede ser instalado en un servidor. De hecho, los componentes DCE son puestos entre las aplicaciones y los servicios de la red, en el cliente y el servidor. Aun cuando DCE es una arquitectura multitarea que contiene un número de servicios básicos, el modelo cliente/servidor DCE esconde detalles de los servicios a los usuarios finales. Esencialmente, los componentes DCE representan un parte integral del modelo de computación, que han sido desarrollados por los gremios de los estándares como ISO (Internacional Standards Organization).

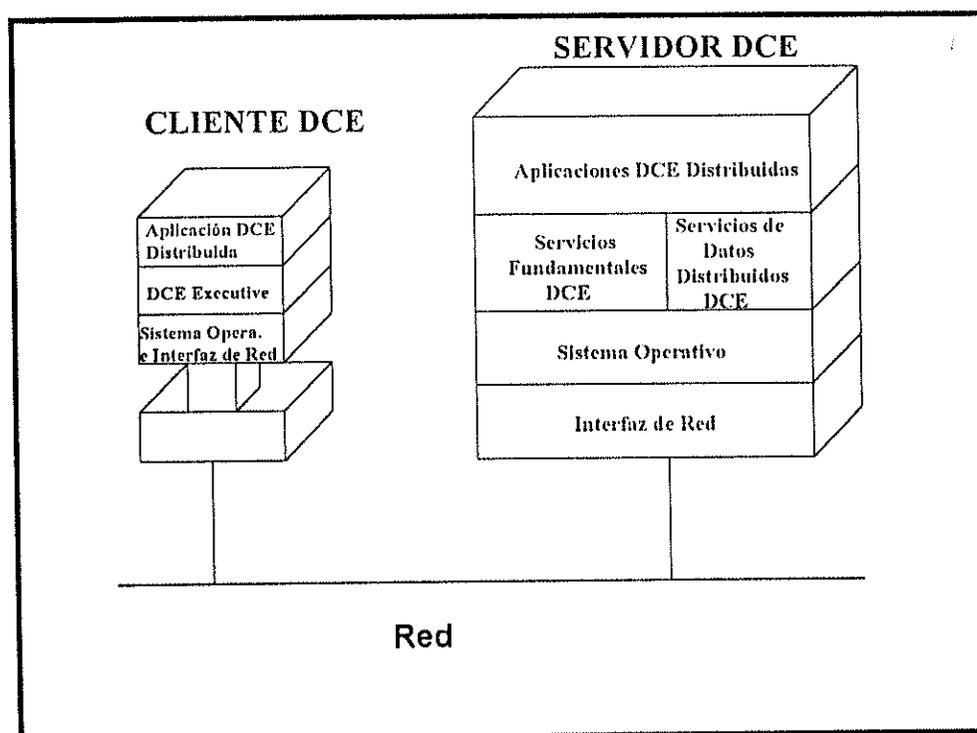


Fig No. 1.6 Modelo Cliente/Servidor DCE.

EL ENTORNO ACTUAL

Los ambientes de computación distribuido representan una contribución invaluable a los desarrolladores de computación distribuida abierta, para hacer de los ambientes de computación distribuida una arquitectura completa. Dos esfuerzos adicionales que han tomado su lugar actualmente son:

- Haciendo las aplicaciones fáciles, disponibles para una gran cantidad de plataformas de hardware y software, proveedores y organizaciones como la OSF (Open Software Foundation) y la UI (Unix International), que están trabajando hacia el desarrollo de un formato de distribución de software de arquitectura neutral (Architecture Neutral Software Distribution Format ANDF). La tecnología ANDF incrementará el atractivo y el poder de los sistemas, por la disponibilidad de un conjunto de aplicaciones ricas, para una gran cantidad de plataformas de computación.
- Así como la computación distribuida se convierte en una realidad, la necesidad de definir un acercamiento consistente para el manejo de sistemas de redes, independientemente del hardware de apoyo y las plataformas de software, se hace cada vez más relevante. La OSF ha propuesto una solución de uniformar la forma de trabajar, la eficiencia, objetos orientados, el manejo efectivo de los costos de los sistemas abiertos para llegar a desarrollar la administración de la distribución.

CAPITULO 2. DESARROLLO DE SISTEMAS DE INFORMACIÓN DISTRIBUIDO

ACERCAMIENTO A LA DISTRIBUCIÓN

La evolución de los ambientes de computación fueron descritos anteriormente, y le han dando vida a varias formas de procesamiento distribuido. La computación distribuida es el siguiente paso evolucionario más allá de compartir archivos.

Los ambientes de computación distribuido, tal como los ambientes de computación distribuida (DCE) de la Fundación de Software Abierto (OSF), hace una colección de sistemas solos (stand alone) y ésta pueda llegar a ser un sistema único. La computación distribuida hace fácil el desarrollo y la ejecución de aplicaciones, que usan recursos a través de la red. Las aplicaciones pueden ser distribuidas, para correr en la computadora que mejor le convenga a la tarea. Varias tareas pueden ser ejecutadas en paralelo proporcionando un gran desempeño y una mejor utilización de los recursos. La arquitectura cliente/servidor es construida sobre el procesamiento distribuido cooperativo, y éste representa un caso especial de la computación distribuida. Hoy en día, los requerimientos cliente/servidor incluyen libertad de almacenar datos y ejecutar aplicaciones en una amplia variedad de plataformas interconectadas. Los distintos modelos cliente/servidor se muestran en la figura No.2.1. Las aplicaciones cliente/servidor deberfan estar disponibles en ambientes de computación de múltiples proveedores, siendo abiertas, flexibles y estandarizadas. La importancia de los sistemas abiertos y los estándares ha sido descrita anteriormente. El siguiente bloque de construcción crítico de la distribución de la arquitectura cliente/servidor son los modelos de ambiente distribuido.

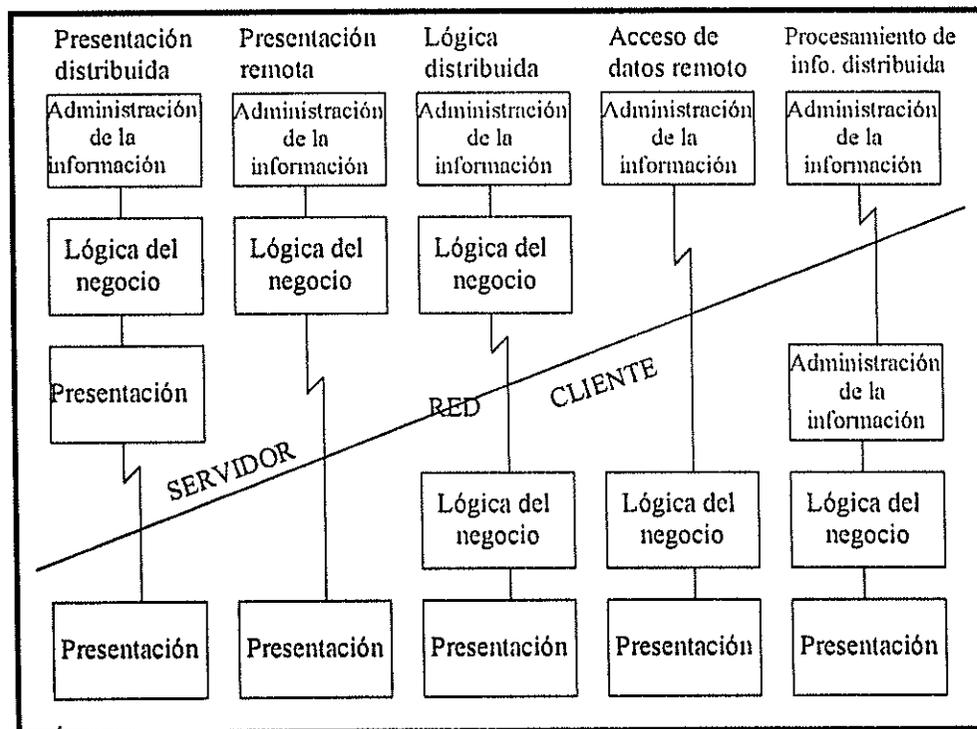


Fig. 2.1 Los modelos Cliente/Servidor.

2.1 MODELOS DISTRIBUIDOS

El desarrollo de la arquitectura de computación distribuida ha sido afectada por dos fuerzas opositoras que prevalecen actualmente en los ambientes de computación. La primera divide (fragmenta) las aplicaciones e impulsa las piezas resultantes (fragmentos) hacia los usuarios finales; existen al menos dos razones para este comportamiento:

- El desarrollo y la ejecución de aplicaciones en estaciones de trabajo y PCs que proveen utilidad precio/rendimiento es significativa.
- Los usuarios finales demandan autonomía local y funcionalidad adicional (interfaces de uso gráfico consistente y flexible) que incrementan la productividad del usuario.

La segunda fuerza opositora tiene sus raíces en la necesidad de los usuarios finales a acceder información corporativa. Esta necesidad afecta los requerimientos de integración y como resultado de la centralización de aplicaciones en grandes y poderosos mainframes. La segunda fuerza incrementa la necesidad de grandes niveles de integridad, rendimiento y disponibilidad.

Esos requerimientos son rara vez alcanzables en sistemas pequeños y simples. La respuesta descansa en el desarrollo de un modelo consistente de distribución de una plataforma centralizada grande, conectado a una red de estaciones de trabajo poderosas, que operan en una red de área local (local area networks o LAN). Las principales interrogantes en esta arquitectura de este modelo son :

- Cómo y dónde los recursos del modelo computacional son distribuidos.
- Cómo implementar las facilidades de intercomunicación entre todos los recursos de computación participantes.

La respuesta a la primera interrogante permite considerar una de las configuraciones más populares actualmente, de los ambientes de computación distribuida al acercamiento multinivel.

2.1.1 AMBIENTE MULTINIVEL

Las organizaciones más grandes han empezado a moverse a lo que conoce como arquitectura de tres niveles (three-tiered architecture), (ver figura No. 2.2). La raíz de la arquitectura de tres niveles es una jerarquía familiar, es la tradicional arquitectura de computación maestra-esclava.

La arquitectura de tres niveles ha agregado capacidades de procesamiento distribuido y cooperativo al modelo de computación jerárquico. Una de las implementaciones más sencillas de esta arquitectura es cuando los recursos de computación están distribuidos verticalmente. Específicamente, el nivel superior es generalmente ocupado por el sistema más poderoso un mainframe, donde reside la información de la corporación. El segundo nivel está conformado por un poderoso servidor LAN, el cual tiene propiedades duales, actúa como el nivel superior de los clientes que envían requerimientos al mainframe; al mismo tiempo ellos funcionan como servidores para las estaciones de trabajo y las PCs que conforman el tercer nivel.

Un ejemplo de las arquitecturas verticales de tres niveles puede ser una organización que extiende su host central (computador central) y las capacidades de su centro de información, al construir LANs en cada uno de sus departamentos y conectando estos servidores LANs al Host.

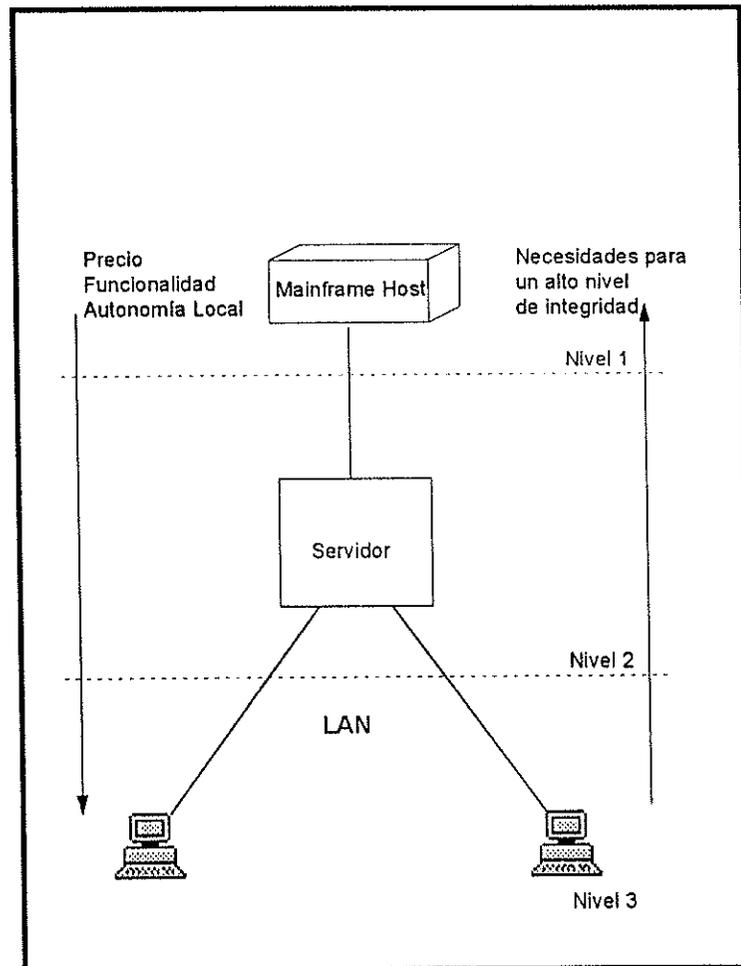


Fig. 2.2 Arquitectura de tres niveles.

Por supuesto, tal arquitectura puede ser expandida horizontalmente, y agregarle mainframes en el nivel superior y servidores de área local en el segundo nivel, (ver figura. No. 2.3). Por ejemplo, una organización que desea extender sus operaciones a otra región, construyendo un segundo centro de información en la nueva región. Puede introducir LANs en las oficinas de venta de cada región, los servidores LANs pueden estar físicamente localizados en áreas remotas (relativo al host), y pueden ser conectados al host a través de redes de área amplia. (Wide Area Network WAN).

La administración de la red, el rendimiento del sistema, la integridad de los datos y la confiabilidad de este modelo extendido de tres niveles, requiere de alguna clase de intercomunicación entre el host del primer nivel y los servidores del segundo nivel. Por esta razón, la arquitectura resultante llega a ser más compleja, a construir y administrar. Sin embargo, la capacidad de computación resultante ha sido incrementada, probablemente en grandes magnitudes, sobre todo la arquitectura resultante es escalable. Esto significa que, así como el número de estaciones crece, adicionalmente las redes de área local, los servidores

más poderosos y aun los niveles adicionales, pueden ser puestos en operación, sin reconstruir la red de trabajo o, más importante para los usuarios finales, sin reconstruir aplicaciones.

Un modelo computacional de varios niveles aparenta ser razonablemente flexible para implementarse. En principio, es una aplicación distribuida cooperativa. En la práctica, sin embargo, para implementar eficientemente la arquitectura multinivel para cualquier aplicación, se deben contestar las siguientes preguntas:

- En qué nivel/es deben ser colocados los datos ?
- En qué nivel/es debe ser colocada la aplicación lógica ?
- En qué nivel/es debe ser colocado una interface de usuario ?

Estas son interrogantes que reflejan las propiedades intrínsecas de la arquitectura cliente/servidor, que son interacciones de los componentes de la aplicación.

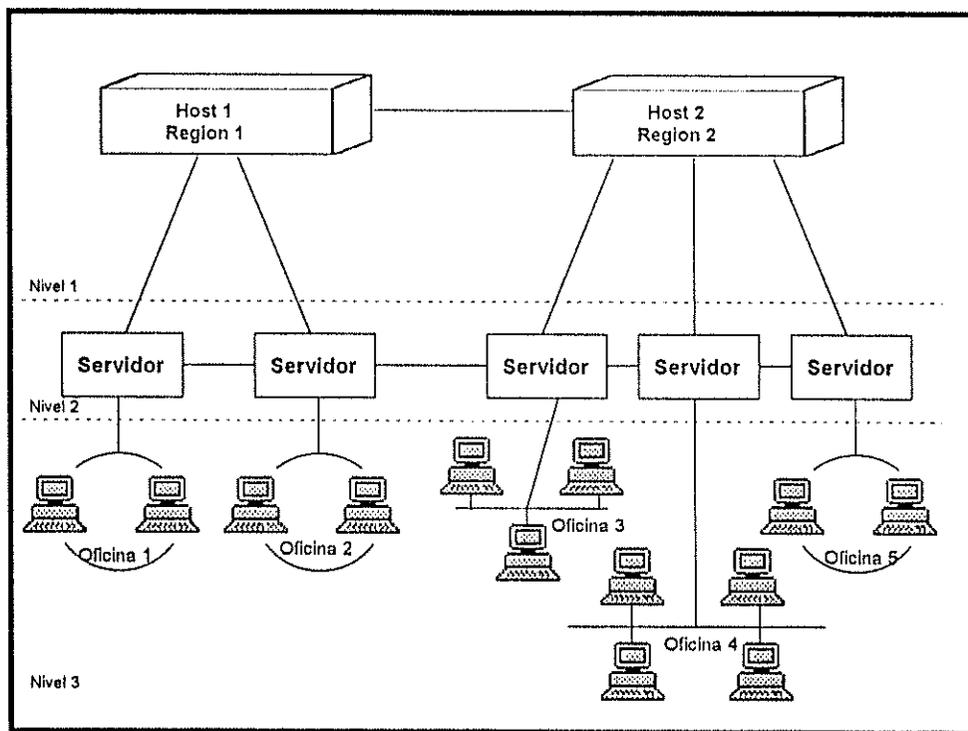


Fig. No. 2.3 Arquitectura de tres niveles extendida.

2.1.2 EL PROCESAMIENTO COOPERATIVO CLIENTE SERVIDOR

Se ha mencionado anteriormente que el procesamiento cooperativo es el fundamento y la fuerza de manejo de la arquitectura cliente/servidor. Las distintas características de las aplicaciones de procesamiento cooperativo, es el alto grado de interacción entre varios componentes de la aplicación (o fragmentos de la aplicación). En la arquitectura cliente/servidor, esas interacciones son las interacciones entre los requerimientos del cliente y las reacciones del servidor a esos requerimientos. Para comprender esas interacciones, veamos los componentes de una aplicación en general. Una aplicación típica consiste de los siguientes componentes (ver figura No. 2.4):

- **LÓGICA DEL PROCESAMIENTO DE PRESENTACIÓN:** ésta es la parte del código de la aplicación que interactúa con los dispositivos, tales como las terminales y estaciones de trabajo de los usuarios finales. La lógica de presentación ejecuta tareas tales como: el formato de la pantalla, la lectura y escritura de información de la pantalla, administración de las ventanas, el teclado y el manejo del mouse⁴. Algunas de estas facilidades que proporciona la lógica del procesamiento de presentación son las interfaces de usuario gráficas (Graphical User Interfaces o GUI).
- **LÓGICA DEL PROCESAMIENTO DEL NEGOCIO:** ésta es la parte del código de la aplicación que usa la información de entrada (de la pantalla y/o base de datos) para ejecutar tareas de negocios. Típicamente, este código está escrito en lenguajes de tercera y cuarta generación.
- **LÓGICA DEL PROCESAMIENTO DE LA BASE DE DATOS:** ésta es la parte del código de la aplicación que manipula la información dentro de la aplicación. La base de datos es administrada por el Sistema de Administración de Bases de Datos (Database Management System o DBMS). La manipulación de la información en Sistemas de Administración de Bases de Datos Relacionales (Relational DBMSs o RDBMS) es hecha usando algún derivado del Lenguaje de Consulta Estructurado (Structured Query Language o SQL). El Lenguaje de Manipulación de los Datos (Data Manipulation Language o DML) de SQL es típicamente encajado en el código de aplicación de tercera (3GL) y cuarta (4GL) generación.
- **PROCESAMIENTO DE LA BASE DE DATOS:** éste es el procesamiento de la información de la base de datos, que es ejecutada por el Sistema de Administración de la Base de Datos (DBMS). Idealmente, el procesamiento del DBMS es transparente a la lógica del negocio de la aplicación; sin embargo, desde un punto de vista de la arquitectura, el procesamiento de la base de datos es una parte esencial de las interacciones del procesamiento cooperativo y debería ser considerada como un componente de procesamiento de la aplicación cooperativa.

⁴ Ratón, hardware utilizado para facilitar el uso rápido del cursor en la pantalla del computador.

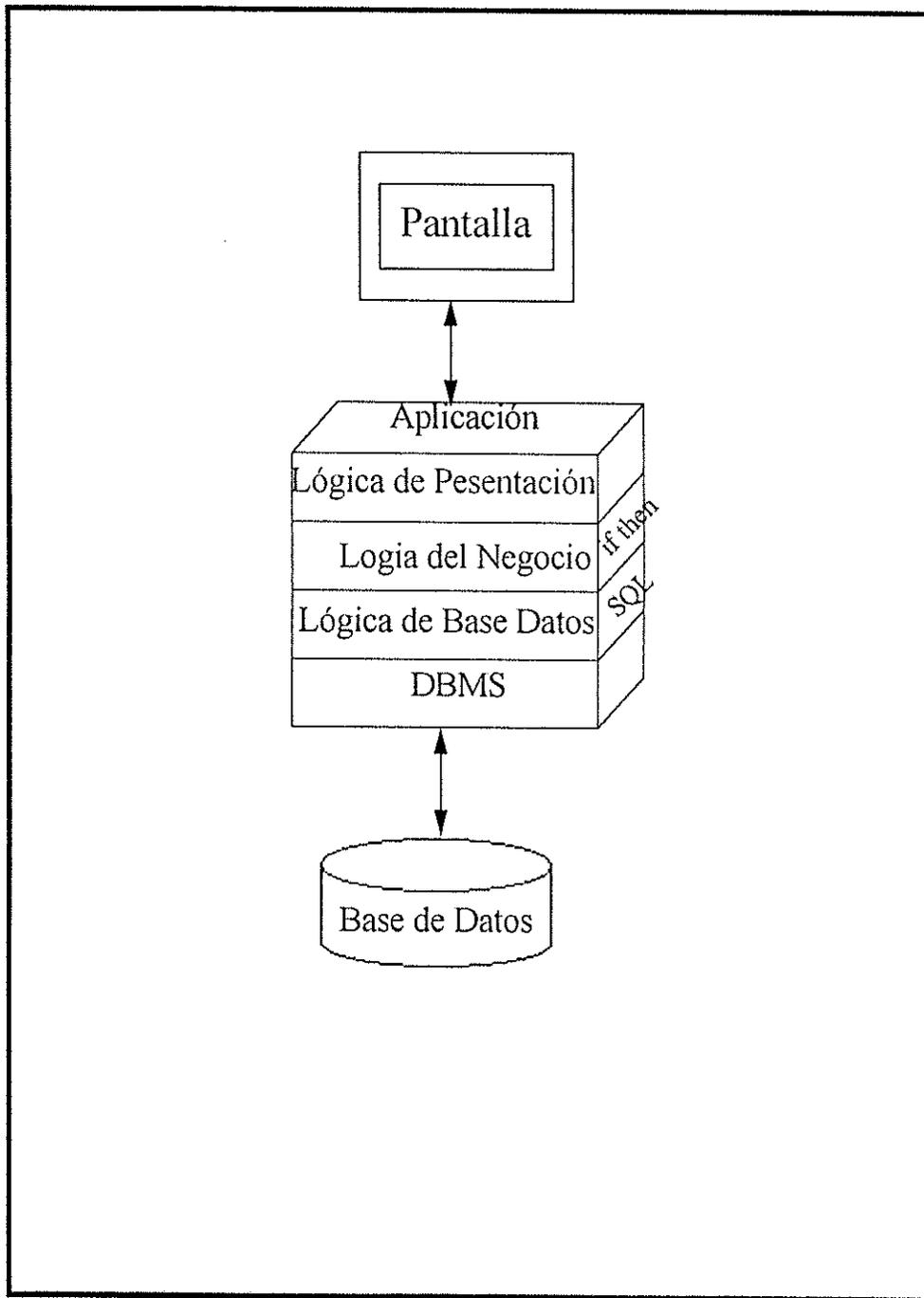


Fig. No. 2.4 Componentes de una aplicación típica.

En el procesamiento basado en un Host, los componentes de esta aplicación residen el mismo sistema y son combinados en un programa ejecutable. La distribución no está teniendo lugar y, en general es restringida por los recursos limitados de la plataforma en la cual es ejecutada. Con la llegada de la computación distribuida, nuevas oportunidades han sido abiertas para los desarrolladores de los sistemas y los usuarios finales; aplicaciones escalables, portables capaces de correr sobre redes de sistemas abiertos que son transparentes a los usuarios finales, pueden ahora ser desarrolladas. Debido a la distribución de los recursos a través de la red, el costo-beneficio puede llegar a ser significativo.

La distribución significa tomar algo, dividirlo en fragmentos y esparcir esos fragmentos. En la computación distribuida, la distribución significa dividir los recursos de computación disponibles en fragmentos y esparcirlos a través de la red. La pregunta es: qué recursos deberían de ser distribuidos y cuáles son las consecuencias de tal distribución?

Cuando solamente los datos son distribuidos en varias localidades, una aplicación sencilla puede hacer concebible el acceso de la información desde cualquier localidad, en un estilo totalmente transparente a la aplicación. Ciertos beneficios (colocar la información cerca de las fuentes, distribución de la información para hacerla altamente disponible) pueden ser derivados de tal distribución. Sin embargo, la singularidad de la aplicación pueden crear un cuello de botella, un factor limitante para llevar a cabo un alto desempeño, portabilidad y el costo-beneficio de las aplicaciones escalables.

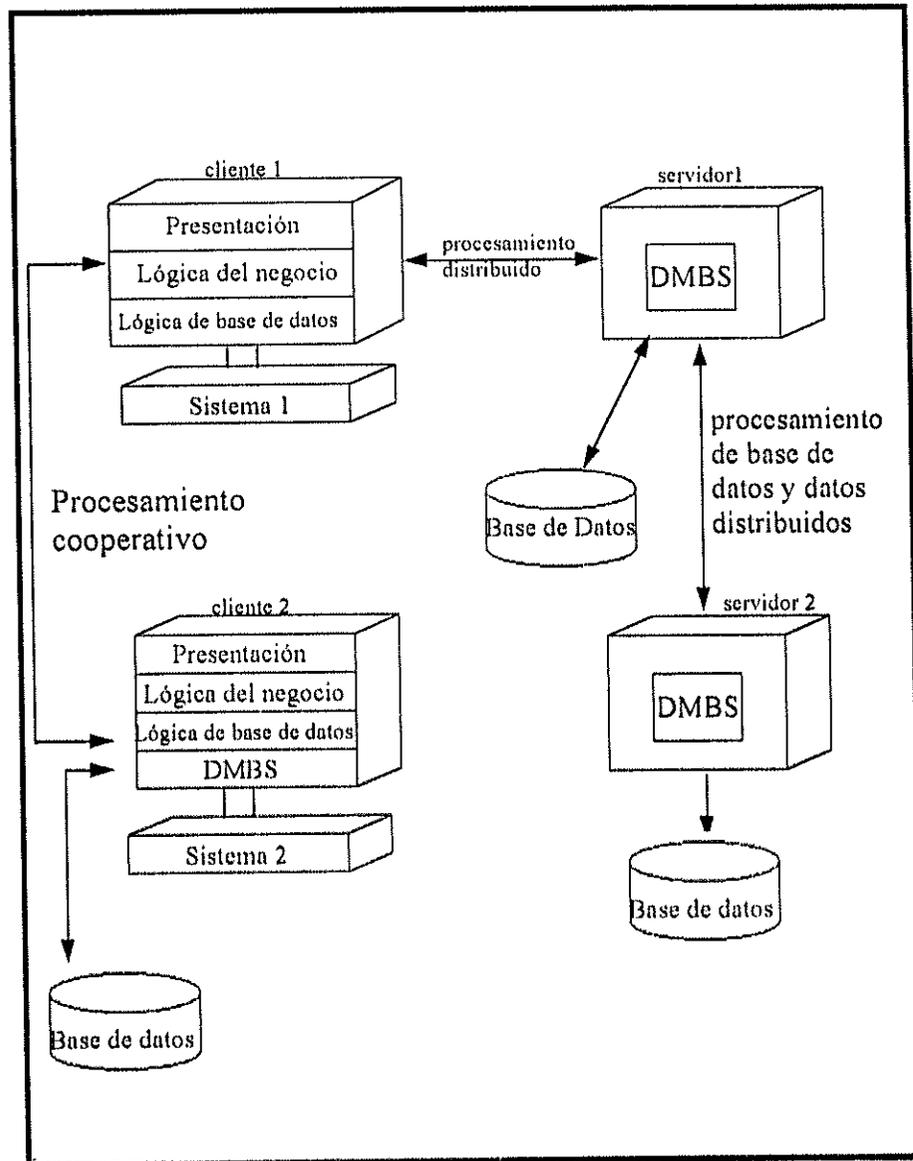


Fig. No. 2.5 Procesamiento Cooperativo y distribuido Cliente/Servidor.

Sin embargo, si en adición a la información, y si alguna aplicación de procesamiento está también distribuida a través de la red, varios de los recursos de computación pueden ser mejor utilizados, considerando especialmente las significativas características de precio-rendimiento de las estaciones de trabajo modernas; por supuesto, una vez los componentes de la aplicación están distribuidos, deben cooperar en el procesamiento de una aplicación de negocios. La arquitectura cliente/servidor emplea procesamiento distribuido cooperativo al:

- Distribuir los componentes del procesamiento de la aplicación entre clientes (presentación y algunas partes de la lógica del negocio) y servidores (algunas partes de la lógica del negocio, la lógica de la base de datos y DMBS).
- Soportando interacciones coherentes entre los clientes y servidores en un estilo cooperativo.

Una de las preguntas que cada diseñador de un sistema cliente/servidor debe de contestarse, es cómo distribuir los componentes de la aplicación, a través de los clientes y servidores. En una arquitectura multinivel, la pregunta sería, cómo distribuir el lugar de los componentes en los distintos niveles. Mientras que no exista una receta universal para la distribución propia de los componentes de la aplicación, algunas recomendaciones generales pueden ser hechas :

- En general, el componente lógico de la presentación, con sus facilidades de entrada/salida, es puesto en el sistema del cliente, y estos clientes son asignados típicamente, en el nivel más bajo del ambiente multinivel (PCs y estaciones de trabajo).
- Dada la disponibilidad de poderosas estaciones de trabajo de los clientes, y de hecho que la lógica de la presentación reside en el sistema del cliente, y tiene sentido colocar además alguna parte de la lógica del negocio en el sistema del cliente. Debería ser así, para al menos esa parte de la lógica de aplicación, que trata con la edición relacionada a la pantalla, y tal vez para aquellas piezas del código que son específicas a un cliente en particular.
- Si la lógica del procesamiento de la base de datos es encajado en la lógica del negocio, y si los clientes mantienen algunas interacciones bajas, información estática, entonces el procesamiento lógico de la base de datos puede ser puesta en el sistema del cliente.
- Dado que una típica red de área local (LAN) conecta a los clientes dentro de un grupo de trabajo de propósitos comunes, y asumiendo que los grupos de trabajo comparten a la base de datos, todas las cosas comunes, fragmentos compartidos de negocios y la lógica del procesamiento de la base de datos y el DBMS debería de ser puesto en el servidor.

Principios similares pueden ser usados para decidir el lugar de los componentes de las aplicaciones cliente/servidor, en ambientes cooperativos multitarea. Para una extensión larga, las preguntas del lugar de los componentes debería ser decidido por :

- La cantidad de información relevante a cualquiera de las aplicaciones.
- El número de usuarios activos corriendo comparado con la información.
- El número de interacciones entre varios componentes de la aplicación.

- Las características técnicas de las plataformas seleccionadas para los clientes y servidores.

Por ejemplo, si la aplicación requiere frecuentemente acceso de escritura y lectura a una base de datos de una corporación, la base de datos y la lógica del procesamiento son puestos en el nivel superior. Si el acceso es menos frecuente y es más de lectura, puede ser factible colocar una copia relevante de la información en el servidor de segundo nivel. El servidor de base del DBMS (generalmente diferente del DBMS del host), la lógica del procesamiento de la base de datos, y las partes compartidas comúnmente de la lógica del negocio, deberían de ser localizadas allí. En este caso, cada nivel contiene su propia base de datos y los diseñadores de sistemas encaran el problema de acceso de datos heterogéneos.

La figura No. 2.6. ilustra una arquitectura donde el nivel superior contiene un mainframe IBM 390, este host accesa la base de datos DB2 de IBM. El segundo nivel es un servidor de base de datos, que corre un DMBS relacional, como un servidor de SQL de Sybase, y el tercer nivel contiene computadoras PS/2 corriendo MS DOS, MS Windows y la base de datos dBase III.

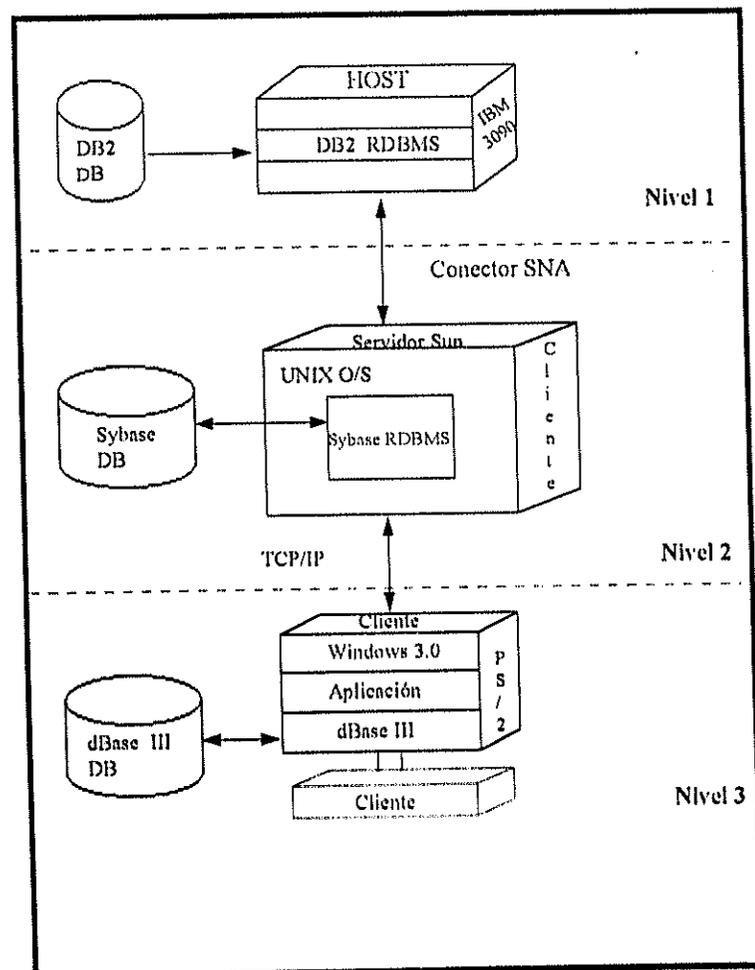


Fig. No. 2.6 Arquitectura de tres niveles con acceso de datos heterogéneo.

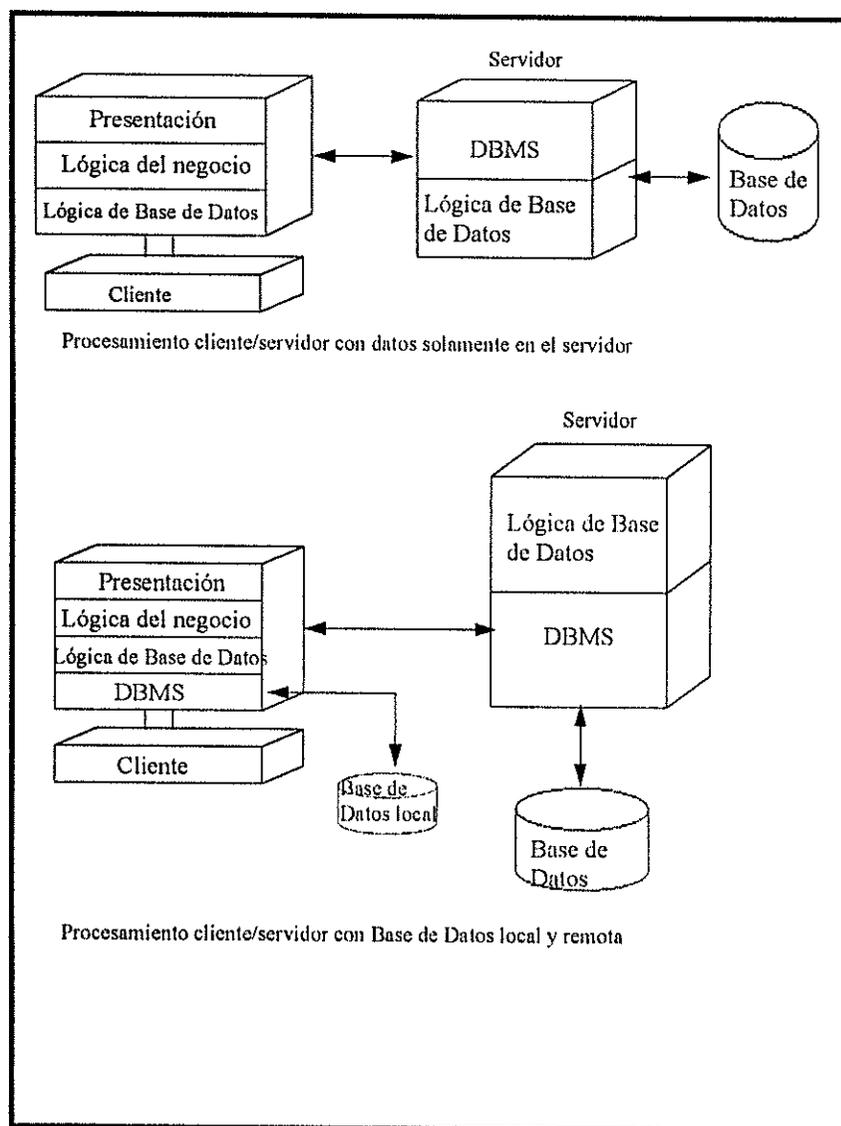


Fig. No. 2.7 Procesamiento Cliente/Servidor.

2.1.3 TÉCNICAS DE PROCESAMIENTO COOPERATIVAS

Para que todos los componentes de la arquitectura ilustrados en la figura No. 2.7 cooperen, ciertas técnicas de comunicación deben de ser empleadas. Teóricamente, hay tres tipos básicos de técnicas de comunicación de procesamiento cooperativo, que la arquitectura cliente/servidor puede utilizar:

- Tuberías (Pipes)
- Llamadas a procedimientos remotos (Remote Procedure Calls (RPC))
- Interacciones SQL cliente/servidor

Los pipes representan un mecanismo orientado a conexión, que pasa información de un proceso a otro. Los pipes son ampliamente utilizados en sistemas basados en UNIX. En principio, el proceso puede estar en diferentes máquinas, aun corriendo sobre diferentes sistemas operativos.

Varias implementaciones de pipes pueden soportar uno o varios mecanismos de transporte concurrentemente. Los detalles de los mecanismos de transporte soportados son escondidos a los usuarios pipes, y éstos imponen un protocolo mínimo y restricciones de formato a los usuarios. Básicamente, los pipes proporcionan facilidades solamente para marcar los límites entre mensajes discretos, para determinar la identidad del remitente, y para efectuar verificación de la recepción del mensaje. Las implementaciones de pipes varían desde la más simple hasta la más compleja arquitectura, como la comunicación avanzada de programa a programa (APPC).

Las llamadas a procedimiento remoto, han sido descritas brevemente en los Ambientes de Computación Distribuido (DEC) de la Fundación de Software Abierto (OSF), Un RPC es un mecanismo por el cual un proceso puede ejecutar otro proceso (subrutina) que reside en un sistema diferente, usualmente remoto, posiblemente corriendo en un sistema operativo diferente. Cualquier parámetro necesitado por la subrutina es pasado entre el proceso original y la subrutina. Como en los pipes, los detalles de mecanismo de transporte usados por los RPC son escondidos al usuario. Una herramienta específica RPC, puede soportar uno o varios mecanismos de transporte diferentes. El principal requerimiento para llevar a cabo una implementación satisfactoria RPC es la habilidad del que llama, a encontrar el servidor donde residen las subrutinas. Una forma de realizar la búsqueda es buscar el nombre de la subrutina requerida con aquellas mantenidas en una base de datos especial de subrutinas/servidores. Las entradas en esta base de datos pueden ser modificadas, de esta manera el servidor donde reside la subrutina, puede ser asignado dinámicamente. Esta asignación puede cambiarse desde la invocación de un RPC a otro.

La interacción SQL cliente/servidor es un mecanismo para pasar los requerimientos SQL y la información asociada de un proceso (usualmente un cliente) a otro proceso (servidor). SQL cliente/servidor es un caso especial de las interacciones cliente/servidor, que son aplicables a las aplicaciones de base de datos relacionales distribuidas. En este caso, el servidor es un servidor de base de datos relacional. Este puede residir en diferentes sistemas (remotos al cliente), posiblemente corriendo sobre diferentes sistemas operativos. La mayoría de los productos implementados cliente/servidor a la fecha, están basados en interacciones SQL cliente/servidor.

Pipes como RPC soportan diversos mecanismos de transporte, los detalles del mecanismo de transporte son escondidos por los desarrolladores de la aplicación, las interacciones SQL cliente/servidor imponen severos protocolos y restricciones de formato a los usuarios. La sintaxis, funcionalidad y los formatos de datos soportados por el SQL son las razones de estas restricciones.

Los pipes, como mecanismos conexión orientada, pueden resolver grandes conjuntos de problemas de intercomunicación en el procesamiento cooperativo. Los RPCs pueden resolver un subconjunto de problemas que resuelven los pipes. Sin embargo, un RPC es un mecanismo sin conexión. Las conexiones RPCs existen solamente en la duración de la llamada. Teóricamente, dos RPC sucesivos a la misma rutina remota, identificada por su nombre, pueden ser ejecutados en diferentes servidores. Las interacciones SQL cliente/servidor, por otro lado representan un mecanismo de conexión orientada similar a los pipes. La conexión aquí es hecha entre el cliente (emisor del requerimiento SQL) y el servidor. Esta conexión es en efecto una conversación cliente/servidor entre compañeros.

La desventaja de las interacciones SQL cliente/servidor es su naturaleza orientada a los datos. De hecho, la arquitectura cliente/servidor que sólo implementa mecanismos de interacción SQL, están limitados solamente a aplicaciones de base de datos relacionales (RDBMS).

Los requerimientos de aplicación, más allá del alcance de una aplicación, no pueden ser satisfechos por las comunicaciones SQL cliente/servidor. Por lo tanto, cualquier sistema abierto cliente/servidor debería ser diseñado para tomar ventaja de varias técnicas de comunicación de procesamiento cooperativo, tal y como se requiere por las aplicaciones lógicas del negocio.

2.2 IMAGEN DE SISTEMA ÚNICO (SSI)

La realización de la arquitectura cliente/servidor toma grandes ventajas de las redes de trabajo, y los avances en precio/rendimiento de la tecnología de microcomputadores, y descarga el procesamiento tradicional mainframe sobre poderosos servidores y estaciones de trabajo. En el ambiente cliente/servidor, la mayoría del procesamiento de las aplicaciones es hecha en el cliente (estación de trabajo), y todas las tareas compartidas comunes, como el acceso a la base de datos, que son ejecutadas sobre el mainframe, o en los servidores de área local. De hecho, ciertas funciones que una aplicación necesita son ejecutadas en diferentes sistemas, potencialmente a miles de kilómetros de distancia de lugar del cliente, no deberían de afectar la manera en que las aplicaciones se comportan. En otras palabras, la distribución de los componentes de las aplicación entre clientes y servidores, no debería cambiar la imagen y sentido de la aplicación. Varios sistemas diferentes pueden participar en la ejecución de un requerimiento sencillo de un cliente. A los usuarios finales el ambiente distribuido completo multicliente/multiservidor, debería de parecer como un sistema poderoso de un usuario único. Para resumir, la implementación propia de un ambiente distribuido cliente/servidor debería proporcionar a los usuarios una imagen de sistema único (Single System Image o SSI).

En términos prácticos, la imagen de sistema único significa que todas las interacciones entre los componentes de la aplicación distribuida (presentación, negocio y la lógica de la base de datos) deberían ejecutarse transparentemente, a tiempo y en forma efectiva. La imagen de sistema único es una característica altamente deseable de la arquitectura cliente/servidor. Sin embargo, los requerimientos de esta implementación son un poco diferentes de la presentación, de la lógica del procesamiento de la aplicación y de la distribución de la información.

2.3 PRESENTACIÓN DISTRIBUIDA

La primera fragmentación de la aplicación que interactúa con los usuarios finales; es la lógica de la presentación. La lógica de la presentación es la ventana de la aplicación hacia el mundo externo. Esta es la lógica que hace la interface con los usuarios finales, y al mismo tiempo, interactúa con la lógica del negocio. En un ambiente basado en un host, la presentación y la lógica del negocio son empaquetadas unificadamente.

Con los avances en la tecnología en las estaciones de trabajo, nuevas capacidades han sido abiertas para los desarrolladores de la lógica de la presentación. Las terminales tradicionalmente basadas en caracteres, están siendo reemplazadas por pantallas de alta resolución, con un número de colores ilimitados. Dispositivos como el mouse, gráficas, imágenes del computador, y aun facilidades de entrada/salida de audio-video están ahora disponibles. Esas innovaciones permitieron a los desarrolladores de aplicación diseñar aplicaciones con interfaces de usuario gráfico (GUI) intuitivas y visuales, que ofrecen características como ventanas, barras de ventanas, mover hacia arriba y abajo los menús y presionar botones. El uso de estas herramientas

y los dispositivos mouse no solamente incrementan la productividad de los usuarios finales, sino también disminuyen y permiten acortar el entrenamiento, para proporcionar un mejor nivel para inversión en el desarrollo de aplicaciones y entrenamiento.

Las razones para la distribución de la presentación son bastante precisas :

- Las gráficas de alta resolución requieren de un significativo poder de procesamiento. Cada punto individual de la figura en la pantalla (pixel) es controlado en su ubicación, intensidad y color, y pueden sobrepasar el millón de píxeles en un monitor de alta resolución.
- Todos los ambientes windows son manejados por eventos. La mayor parte del procesamiento está controlado por el apuntador del mouse, que se mueve de una ventana a otra al presionar el botón del mouse, o cuando un ítem del menú es tocado. Este procesamiento requiere significativos recursos de computación.
- Las estaciones de trabajo frecuentemente diseñadas para ejecutar elevadas funciones de gráficas. Los procesadores están mejorados para el manejo de gráficas de 2-D y 3-D (dimensiones).

Los requerimientos de poder computacionales, para la mayoría de las interfaces de usuario gráficas actualmente usadas, crean un beneficio económico, al liberar la lógica de la presentación en una plataforma que está diseñada específicamente para manejar dichos requerimientos; por la naturaleza de la presentación, tal plataforma está diseñada para las interacciones humanas y debería estar en la estación de trabajo del usuario. Para satisfacer los requerimientos de imagen de sistema único, las interacciones entre los fragmentos de la presentación y el resto de la lógica de la aplicación, deberían estar complementados en una forma transparente para el usuario.

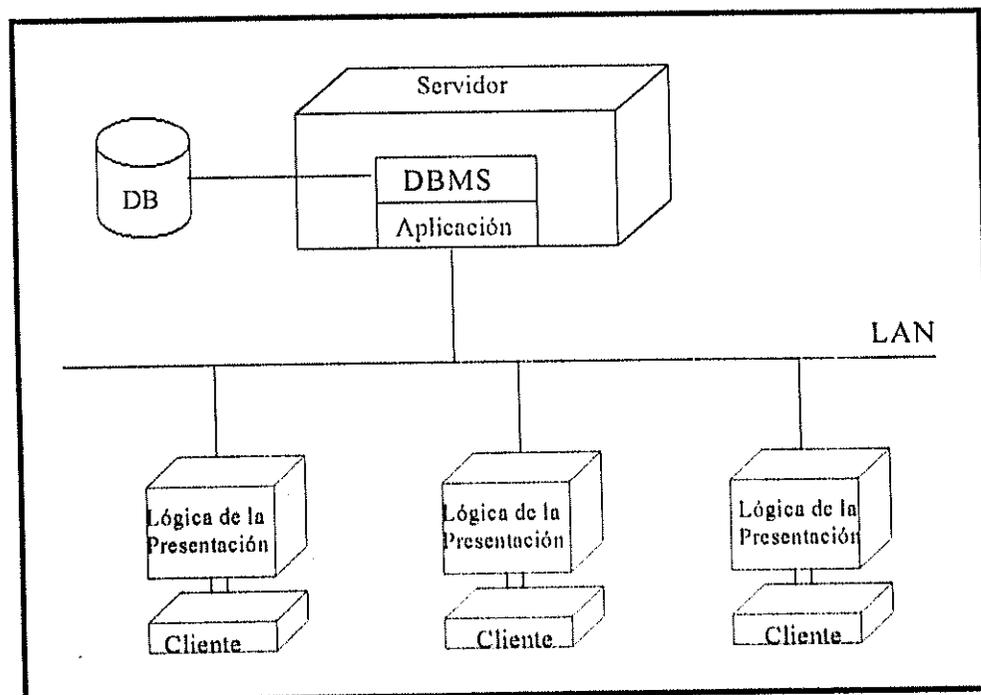


Fig. No. 2.8 Presentación distribuida.

2.4 PROCESAMIENTO DISTRIBUIDO

Mientras la lógica de la aplicación es la cara de la aplicación a las interacciones del usuario, ésta realmente representa la lógica del negocio y de la aplicación, que es la esencia de la aplicación. En el ambiente cliente/servidor, la lógica de la presentación es típicamente localizada en el sistema del cliente. Una de las interrogantes críticas de diseño en este punto es la ubicación de la lógica del negocio y la base de datos (refiriéndose a ambas, como el procesamiento de la aplicación). Obviamente, hay tres posibles soluciones :

- La lógica del procesamiento de la aplicación es ubicada totalmente en el sistema del cliente.
- La lógica del procesamiento de la aplicación está ubicada totalmente en el servidor.
- La lógica del procesamiento de la aplicación está fragmentada, con fragmentos distribuidos entre el cliente y el servidor.

Cada solución de éstas tiene sus ventajas y desventajas :

CASO 1. El procesamiento de la aplicación reside en el sistema del cliente. Hay varias razones para ubicar la lógica del procesamiento de la aplicación en el sistema del cliente; algunas de éstas son :

- El precio/rendimiento de las últimas estaciones de trabajo es de un orden de magnitud más elevado que el de los mainframes tradicionales. Esto es económicamente conveniente para descargar lo caro del ciclo de las máquinas mainframes, comparado con el bajo precio de las microcomputadoras. Recuerde que en la arquitectura de tres niveles, el segundo nivel está compuesto de servidores LANs basados en microcomputadores, que actúan como servidores para las estaciones de trabajo de los clientes del tercer nivel y como clientes de los mainframes del primer nivel.
- Asumiendo que al menos una porción de la lógica del procesamiento de la aplicación trata con terminales E/S, ejecutadas por la lógica de la presentación, tiene sentido ubicar la lógica del procesamiento de la aplicación tan cerca de la fuente de la terminal de datos E/S como sea posible. La alternativa es enviar la información de la pantalla para procesarla y luego enviar los resultados sobre la red.
- Si toda la lógica del procesamiento de la aplicación reside en el sistema del cliente, el tráfico de la red entre la presentación y la lógica del procesamiento de la aplicación, puede ser reducido significativamente.
- No es necesario la sincronización entre los fragmentos de la lógica de la aplicación, la cual sería necesaria si los fragmentos están distribuidos entre clientes y servidores.

Algunas desventajas de llevarla a cabo son :

- La necesidad de mantener múltiples copias de la misma lógica del negocio en cada estación de trabajo del cliente representa serios problemas de mantenimiento.
- Los beneficios del código modular común compartido no puede ser realizado.

- El poder de la máquina servidora no es completamente elevado. Al mismo tiempo, el sistema del cliente podría ser sobrecargado, el cual podría necesitar un cambio de hardware e inversión adicional en múltiples estaciones de trabajo.

CASO 2. La lógica de procesamiento de la aplicación reside en un sistema servidor. Hay varias razones para ubicar la lógica del procesamiento de la aplicación en el sistema servidor; algunas de ellas son :

- Al ubicar los módulos comunes de la lógica del negocio en un sistema central se elimina el código redundante, se simplifica el mantenimiento del sistema, y permite un mejoramiento en la inversión en el hardware del servidor.
- Al Ubicar la lógica del negocio en el servidor de base de datos, se reduce el tráfico de la red entre la aplicación y la base de datos.
- Aquí no hay necesidad de la sincronización entre los fragmentos de la lógica del procesamiento de la aplicación, que es necesaria si los fragmentos están distribuidos entre los clientes y servidores.

Algunas desventajas de llevar a cabo es :

- Al ubicar completamente la lógica del procesamiento de la aplicación sobre el sistema servidor, puede resultar en una sobrecarga de los recursos del servidor, y se puede necesitar un cambio de hardware. Esto es realmente verdadero cuando diferentes aplicaciones tienen que compartir el mismo servidor.
- Las interacciones, entre la lógica de la presentación y el negocio, pueden aumentar significativamente el tráfico de la red y el tiempo de respuesta de las aplicaciones.
- El poder de las estaciones de trabajo puede ser subutilizado reduciendo el retorno de la inversión.

CASO 3. La lógica del procesamiento de la aplicación es distribuida entre los sistemas del cliente y servidores. Esta aproximación puede combinar las ventajas del primero y segundo caso. Mientras se reducen las desventajas de los otros casos, de hecho fragmentos comunes de la lógica de la aplicación pueden ubicarse en el servidor del sistema, cerca de la base de datos. Simultáneamente, los fragmentos relacionados a la presentación individual pueden ser localizados cerca de la plataforma de presentación (sistema del cliente). Las redundancias pueden ser eliminadas y el tráfico de la red puede ser optimizado, y el poder computacional de las estaciones de trabajo y el servidor pueden ser elevados. Es posible y deseable fragmentar la lógica del procesamiento de la aplicación, tanto que el procesamiento que reside en el servidor y el clientes, puedan ser ejecutados simultáneamente en paralelo, y el resultado en todo el tiempo de computación es reducido.

Al parecer, la aproximación a la distribución provee mayores beneficios. Esto también puede ser más complicado. La fragmentación particular es muy importante para el éxito de la distribución. El balance correcto entre la lógica del procesamiento de la aplicación, el procesamiento residente en el cliente y servidor depende de la aplicación.

La distribución de la lógica del procesamiento de la aplicación es una de las tareas de diseño críticas de las aplicaciones cliente/servidor. Hoy día, esto puede ser solventado por el uso de nuevas y emergentes metodologías de diseño de aplicaciones distribuidas.

Otro factor crítico de la distribución de la lógica de la aplicación es el éxito de la sincronización en el procesamiento transaccional distribuido. El fragmento de la lógica de la aplicación residente en el cliente y el fragmento del servidor, interactúan en forma cooperativa intercambiándose requerimientos y respuestas. En general, la interacción en procesamiento transaccional distribuido entre clientes y servidores es sincrónica. Un cliente emite un requerimiento al servidor y espera a que el requerimiento sea procesado. El resultado positivo o negativo es devuelto al cliente antes del siguiente requerimiento.

El problema surge cuando en el curso de la transacción, tanto el cliente como el servidor modifican sus respectivos recursos. Si ambos han tenido éxito en su parte del proceso de transacción, pueden realizar los cambios a los recursos de transacción haciendo entonces los cambios permanentes. Si una de las interacciones compañeras cliente/servidor (el proceso del cliente o el proceso del servidor) falla, el otro compañero debería de revertir los cambios hechos en el transcurso de la transacción, al estado original de los recursos, antes de que la transacción fuera comenzada.

Esta reglas son llamadas protocolo de realización de dos fases (two phase commit). Aun cuando éstas son frecuentemente asociadas con el procesamiento de bases de datos distribuidas, el protocolo de realización de dos fases es el resultado del procesamiento transaccional distribuido. El protocolo de realización de dos fases garantiza la integridad de los recursos de la transacción.

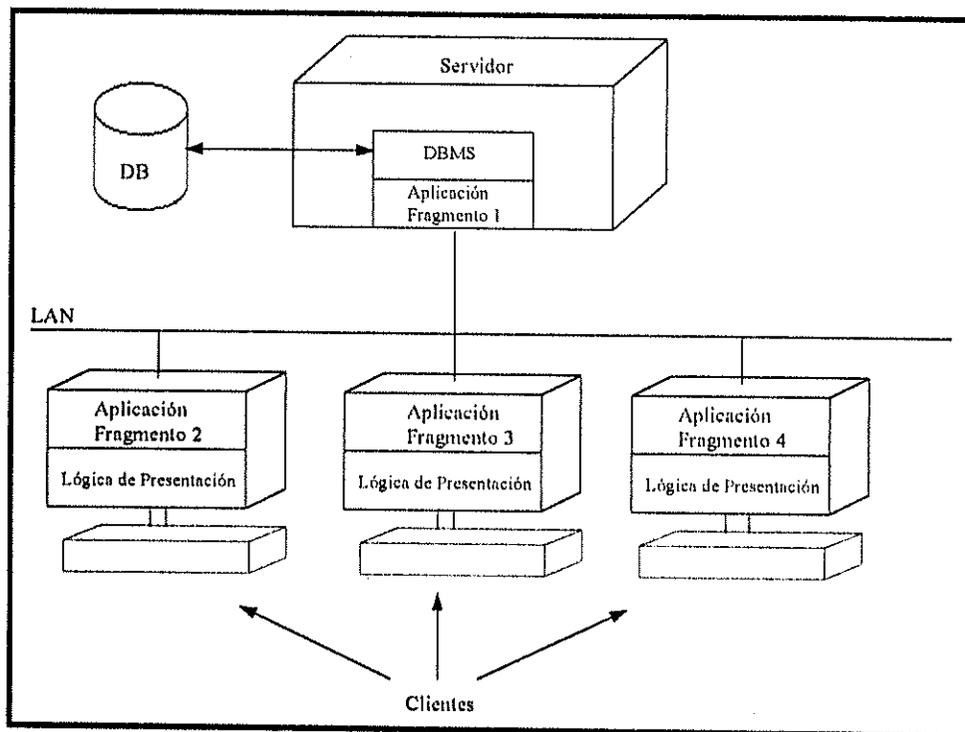


Fig. No. 2.9 Procesamiento distribuido.

2.5 INFORMACIÓN DISTRIBUIDA

Hasta ahora, el análisis de la distribución de los componentes de la aplicación, se ha concentrado en la lógica de la presentación, negocios y bases de datos; cada uno de estos componentes, sin embargo, tiene relación con los datos. La lógica de base de datos accesa a la información desde un DBMS; la lógica de negocios procesa los datos, y la lógica de presentación despliega la información a los usuarios finales. De hecho, el procesamiento de datos es el propósito principal de una aplicación. Hay una serie de interrogantes vitales relacionadas con el logro de la distribución de los datos en un ambiente cliente/servidor; algunas de esas preguntas son :

- Es la información distribuida o centralizada ?
- Qué papel juega un servidor de base de datos ?
- Dónde se localiza la información, si es que está distribuida ?
- De qué manera está la información fragmentada, si es que está distribuida ?
- La información está replicada en varios lugares ? De ser así, de que manera se mantienen copias actualizadas ?
- Cómo pueden ser accedados los datos en múltiples localizaciones, en una forma de aplicación transparente ?
- Cómo se puede garantizar la integridad y disponibilidad de los datos ?
- Cuál es el flujo de la administración de la información en un ambiente distribuido ?

Estas preguntas y otros aspectos relacionados con la información son extremadamente importantes para entender la arquitectura cliente/servidor. Sin embargo, la pregunta acerca de que si los datos se localizan en el sistema del cliente o del servidor, puede ser contestada ahora.

Conexiones directas e indirectas. Así como la arquitectura cliente/servidor evolucionó del procesamiento de dispositivo compartido, uno de sus principales objetivos fue permitir a una aplicación basada en un cliente acceder información remota eficientemente. Este acceso es proporcionado por un servidor de base de datos. Un servidor de base de datos contiene el software DBMS y los datos por sí mismos (base de datos) :

- El software DBMS, por su naturaleza, está diseñado para ser común y compartido.
- En un ambiente cliente/servidor, el servidor es el punto focal de todos los requerimientos del cliente.
- Las implementaciones avanzadas DBMS permiten la ubicación de procedimientos comunes y hasta de ciertas reglas del negocio en el servidor DBMS; este servidor DBMS necesita una posición central en una arquitectura cliente/servidor.

- En un ambiente de trabajo de grupo, la mayoría de la información que va a ser procesada necesita estar compartida entre todos los clientes.
- Las implementaciones avanzadas DBMS incluye diccionarios de datos DBMS residentes, que facilitan el desarrollo de la aplicación, brindan soporte a la transparencia de localización de los datos y proveen una administración de datos más eficiente.
- El colocar los datos y el DBMS juntos en un servidor de base de datos facilita la implementación que proporcionan integridad y disponibilidad de los datos.

Por lo tanto, la decisión arquitectónica apropiada sería colocar el software DBMS y todos los datos compartidos en un servidor de base de datos. Al mismo tiempo, si las aplicaciones del cliente requirieran, algún dato único y específico con relación a las aplicaciones y el cliente, el lugar apropiado para ello sería un almacenador de datos localizado en el sistema del cliente. Estas dos situaciones determinarían la manera en que la aplicación del cliente se podría conectar al servidor corriendo un DBMS. Con la conexión de la aplicación al DBMS directa, la aplicación del cliente utilizaría protocolos de comunicación y algún tipo de procesamiento de transacción remota, para requerir al servidor DBMS residente que accese la información. Esta situación se puede complicar más, si los múltiples servidores deben ser accedidos.

- Una aplicación podría utilizar procesamiento de transacción distribuida para establecer sesiones con múltiples servidores.
- Uno de estos servidores puede propagar las solicitudes de los clientes a servidores adicionales. Tal proceso puede estar basado en nombres de servidores remotos, y el servidor original puede actuar como un coordinador de una transacción distribuida, o bien enviar sus requerimientos a un coordinador designado.

En cualquier caso, la integridad de los datos y los problemas de administración tendrían que ser resueltos, por ejemplo, si los datos de aplicación se encuentran dispersos en múltiples servidores (horizontalmente entre niveles o verticalmente entre un servidor LAN y un host); la administración de base de datos distribuidos o el procesamiento de requerimientos distribuidos con la realización de las dos fases, será requerida para garantizar la integridad de los datos a través de los servidores. Una imagen de sistema único, puede ser implementada apoyando la transparencia de localización de los datos independientemente del servidor en el cual residan los datos.

Con una conexión de aplicación al DBMS indirecta, el cliente solicita información remota a través de un intermediario local DBMS (cliente residente). El beneficio de tal conexión indirecta es que el cliente puede acceder tanto a los datos remotos, como a los locales, usando el mismo lenguaje de manipulación de información.

Un problema con la conexión indirecta es que, desde un punto de vista de aplicación, la información es distribuida por lo menos entre el cliente y el servidor. Por lo tanto, de manera similar a la conexión directa, la integridad de la información distribuida puede ser garantizada por una administración de transacción distribuida o un procesamiento de requerimiento distribuido con la realización de dos fases. Asimismo, la transparencia de ubicación de los datos debería estar soportada para proveer la imagen de sistema individual.

Otro problema tiene lugar cuando el cliente y el servidor DBMS son productos diferentes; éste es el problema de acceso heterogéneo DBMS. Las diferencias, en los dialectos de los lenguajes de manipulación de la información, podrían necesitar el desarrollo de pasarelas (gateways) DBMS a DMBS.

Es por ello que tanto la conexión directa e indirecta, entre aplicaciones y DBMSs poseen problemas de datos distribuidos similares. La diferencia principal reside en la asignación de responsabilidades para resolver estos problemas. En el caso de la conexión directa, un servidor DBMS residente debería manejar los requerimientos de información, mientras que para la conexión indirecta, los requerimientos de datos deberían estar distribuidos entre el cliente DBMS y el servidor DBMS, ver figuras 2.10 y 2.11.

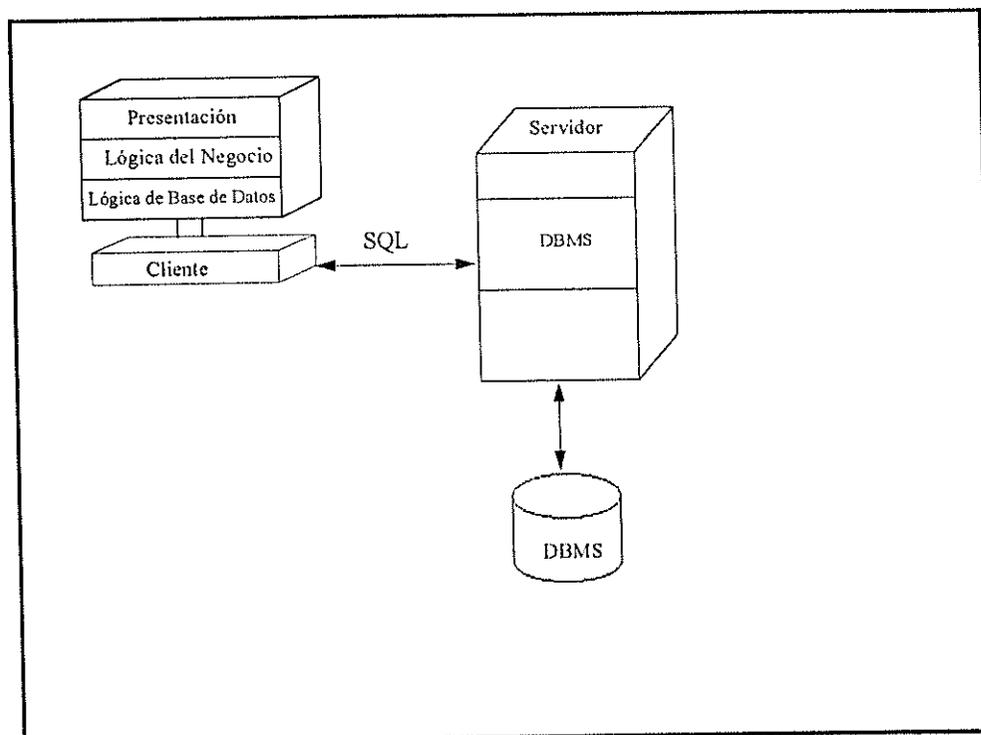


Fig. No. 2.10 Conexión Directa.

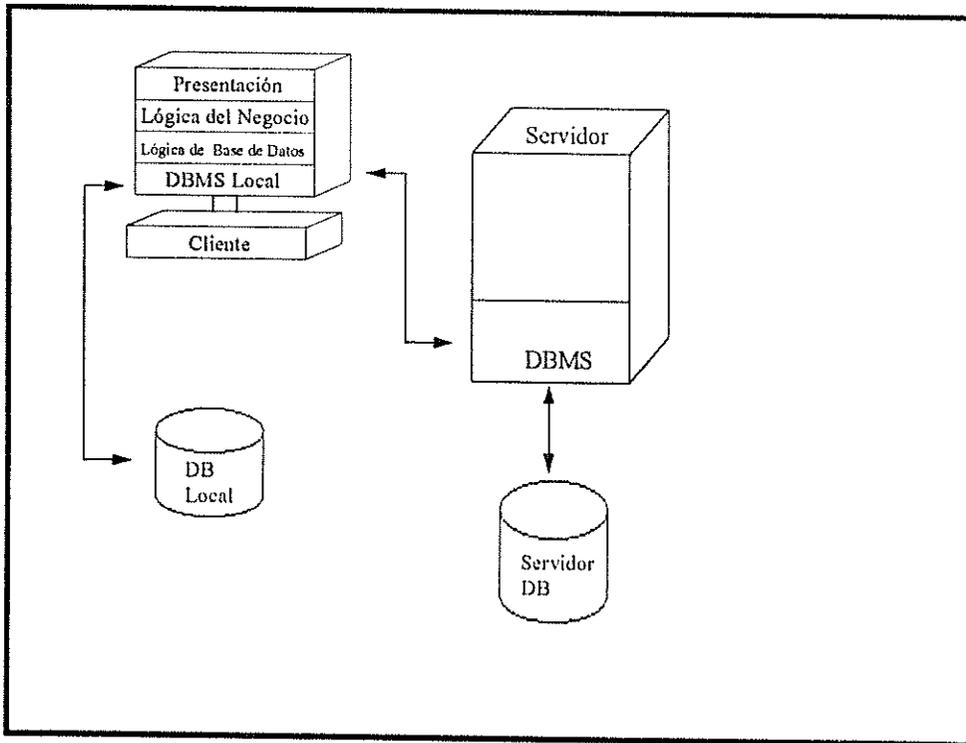


Fig. No. 2.11 Conexión Indirecta.

Para ilustrar los problemas potenciales con la distribución de datos en un ambiente cliente/servidor distribuido de tres niveles, considere el siguiente escenario: Una compañía de seguros de vida estructura su unidad de pólizas individuales, como una organización de tres niveles de la siguiente manera:

- La oficina central, que es la que mantiene todos los libros corporativos y registros financieros.
- Los centros de servicio regional, que controlan todas las transacciones de servicio al cliente para una región en particular.
- Y agentes de seguros quienes, dan servicio a un número de sus clientes.

Para mejorar el servicio al cliente y reducir los gastos operacionales en los centros de servicio regional, la compañía decide equipar agentes con computadores personales portátiles (laptop), capaces de ejecutar varias aplicaciones de seguros típicas. Periódicamente, los agentes deberfan conectarse a las computadoras (servidores) en los centros de servicio regional para descargar las transacciones (nuevos clientes, cambios en las pólizas etc.), y adquirir información actual de la oficina central y de las computadoras del centro de servicio regional. El requerimiento de negocio para la compañía es que ningún cambio es válido hasta que los libros corporativos estén al día.

Arquitectónicamente, este escenario es un ambiente cliente/servidor, con agentes con computadoras portátiles en el tercer nivel, servidores del centro del servicio regional como un segundo nivel, y un mainframe de la oficina central como el primer nivel. Sin embargo, para una implementación exitosa todos los problemas relacionados con el cliente/servidor mencionados anteriormente necesitan ser resueltos.

- Los agentes llevan sus computadoras personales (laptop) con sus clientes, y corren las aplicaciones apropiadas. Entonces, al menos alguno de los datos del cliente debe residir en la computadora personal del agente. Las PCs laptops están corriendo el sistema operativo MS DOS y bases de datos DBASE III.
- Los servidores del centro de servicio regional deben de manejar toda la información relacionada con los agentes asignados, a aquel centro de servicio regional, más algunos datos adicionales para un sistema de manejo de soporte de decisiones de la administración del centro de servicio regional. En concordancia con la visión de sistemas abiertos y con el fin de alcanzar un precio/rendimiento máximo, los servidores del centro de servicio regional están corriendo un sistema operativo UNIX y un sistema de administración de bases de datos relacional SYBASE.
- La oficina central mantiene todos los libros corporativos y todos aquellos registros relevantes de los clientes. El mainframe de la oficina central está corriendo sistemas de administración de base de datos IBM MVS/ESA y DBW e IMS.
- Las actualizaciones efectuadas por los agentes, a los datos en DBASE III, deben ser propagadas al servidor de centros de servicios regional, y a las bases de datos de la oficina central DB2 e IMS. Para algunas transacciones, la integridad de los datos es obligatoria; si la propagación fracasa en cualquier nivel, todos los cambios deben de ser revertidos desde las bases de datos en otros niveles.
- Para alcanzar la portabilidad de la aplicación, las aplicaciones de un agente deben ser designadas de tal manera, que los datos requeridos puedan ser accedados transparentemente de su localización o del DMBS en el cual descansa. Cada DBMS involucrado en este ambiente usa su propio lenguaje de manipulación de datos, dos DBMS (DBASE III e IMS) no están basados en SQL (es decir no son relacionales).

Este ejemplo demuestra algunos de los problemas que un diseñador de ambiente cliente/servidor debe resolver. Respecto a los problemas de datos distribuidos, algunos de ellos y los correspondientes requerimientos fueron formulados por C.J. Date, uno de los primeros diseñadores de bases de datos relacionales.

2.6 DISTRIBUCIÓN Y ADMINISTRACIÓN DEL SOFTWARE

Una vez implementada la arquitectura cliente/servidor distribuida, los problemas de distribución y administración del software toman un lugar extremadamente importante. De hecho, existen problemas de redes, administración de datos y problemas de administración, así como problemas de distribución de información de los cuales se habla muy poco. Los problemas más recientes son muy importantes para la implementación exitosa de la arquitectura cliente/servidor. Considere que un agente PC está corriendo la lógica del software de presentación (Microsoft Windows), lógica de aplicación, y lógica de base de datos (DBASE III) proporcionada por la compañía. Debería alguno de estos componentes del software cambiar (debido a la nueva versión de software adquirido, cambios en las aplicaciones desarrolladas internamente, o cambios en las regulaciones de los negocios) Los agentes esperan tener el software de su PC actualizado. Como debería ejecutarse esta actualización? Debería la oficina central enviar un conjunto de disquetes flexibles a cada uno de sus agentes? Debería cada agencia retornar su PC al centro de servicio regional para darle mantenimiento?, etc.

CAPITULO 3

ESPECIALIZACION DEL CLIENTE Y DEL SERVIDOR EN AMBIENTES DISTRIBUIDOS

3.1 CLIENTES

Ya que los nodos de los clientes están diseñados típicamente para interactuar con los usuarios finales, su funcionalidad e implementación puede ser especializada por estas interacciones. Similar a desarrollar células sensoriales especializadas en humanos y animales, los nodos del cliente deben ser diseñados para desempeñar funciones tales como: gráficas de alta resolución y sonido a un nivel de desempeño aceptable. Sin embargo, los requerimientos de aplicación aceptable muchas veces exceden aquellos de las interacciones de los nodos de cliente, y pueden ser satisfechos sólo como resultados del desempeño colectivo de todos los nodos del sistema.

3.1.1 EL ROL Y LAS FUNCIONES DE LOS CLIENTES

La mayor parte de las funciones ejecutadas en un sistema de clientes en un ambiente cliente/servidor, son las funciones de presentación y algunas lógicas de negocios. Las interacciones del usuario final con una aplicación, son ejecutadas a través de la lógica de presentación. La lógica de presentación es la capa de aplicación que por un lado interactúa con la lógica de la aplicación y, por el otro lado, interactúa con los usuarios finales. La lógica de presentación más reciente incluye todas las interacciones con el dispositivo físico (terminal) y el manejo de entradas/salidas ejecutadas por el usuario final (E/S pantalla, E/S teclado, E/S mouse, etc.).

Las funciones de presentación tradicional distribuidas con presentación basada en ambiente caracter, donde el procesador despliega secuencialmente caracteres recibidos de una aplicación en la pantalla. La evolución continua de las funciones de presentación ha estado muy relacionada con el alto desempeño de las estaciones de trabajo y ofrecen capacidades de despliegue gráfico.

Estos despliegues gráficos permiten al procesador controlar los píxeles⁵ en la pantalla, y construyen un mapa de bits especial. Cada bit en el mapa corresponde a un pixel en la pantalla, y el procesador puede pintar cualquier figura, sea este caracter, símbolo, o dibujo; la resolución de la pantalla está definida por el número de píxeles disponibles. Por lo tanto, el color, tamaño, y la ubicación de los objetos en una pantalla, no está limitada por un font⁶ de caracter en particular, o por el número de filas y columnas. Estas capacidades de despliegue permiten, a los diseñadores de software, crear interfaces de aplicación únicos, efectivos, e intuitivos, que utilizan facilidades de gráficas de imagen en la computadora, y hasta de E/S de audio/video; por ejemplo, una aplicación de oficina que despliega varias figuras (llamadas iconos): Un gabinete de archivo para indicar el almacenamiento de archivo, una figura de un sobre para un documento individual. El usuario final puede apuntar y presionar el mouse en un documento, y el texto del documento se despliega en la pantalla. Hasta las aplicaciones basadas en ambientes caracteres tradicionales, pueden utilizar el poder del despliegue gráfico sin cambios en las aplicaciones.

⁵ Pixel : punto en la pantalla que tiene determinados atributos, como color, intensidad.

⁶ Tipo de caracter

Obviamente, el sistema del cliente que tiene estas capacidades de despliegue, debe especializarse en las presentaciones gráficas. La lógica de aplicación que utiliza correo electrónico, y software de control de acceso a los recursos del servidor, por ejemplo, puede ser desarrollado separadamente de la lógica de presentación. Generalmente, esto no sucede en una especialización de estación de trabajo del cliente.

3.1.2 ADMINISTRACIÓN DE LA PRESENTACIÓN

La especialización de los sistemas de cliente, para la administración de la presentación, debe conformar las metas comunes de la arquitectura cliente/servidor en un ambiente distribuido abierto. La administración de la presentación debería estar basada en estándares, que permitan a los sistemas interactuar con los usuarios en un estilo consistente; deberían ser portátiles a través de una amplia variedad de plataformas de hardware, y ser capaces de interoperar con otras aplicaciones de sistemas abiertos.

PORQUÉ UN INTERFACE DE USUARIO GRÁFICO ESTÁNDAR

Las interfaces consistentes entre usuarios y aplicaciones, representan un requerimiento clave para los sistemas abiertos. Este requerimiento es extremadamente importante, ya que las interfaces de usuario afectan tanto a quienes los desarrollan como a los usuarios finales. De hecho, mientras algunos programas o interfaces de bases de datos, se esconden de los usuarios, la presentación de la aplicación es visible a cualquier usuario, y aun al usuario casual del sistema. Las interfaces de presentación entre usuarios y aplicaciones reciben el nombre de interfaces de usuario gráfica (GUI), y están diseñados para presentar la información a los usuarios en una forma gráfica. Las gráficas no necesariamente significan figuras, los procesadores de palabras o el software con múltiples clases de textos, tamaños y estilos también requieren una presentación gráfica.

La amplia variedad de interfaces disponibles actualmente (incluyendo aquellos basados en caracteres), pueden confundir a usuarios y desarrolladores. Cada nuevo interface requiere que los usuarios y desarrolladores sean reentrenados, y las aplicaciones sean modificadas. Un nuevo interface de usuario gráfico puede causar que la aplicación entera sea reescrita, para una nueva plataforma que soporte el interface de usuario gráfica deseado. Típicamente, las aplicaciones escritas para un interface de usuario gráfico, no pueden ser transportadas hacia otros ambientes de interface gráfica. Muchos desarrolladores deben de tomar decisiones difíciles y costosas, y limitar el desarrollo del producto a interfaces particulares.

Por ejemplo, interfaces incompatibles desarrollados para Macintosh, Microsoft Windows, y el administrador de aplicación OS/2. Por lo tanto, un interface de usuario gráfico estándar (GUI) con un API (interface de programación de aplicación o Application Programming Interface) común individual, y una imagen y sentido estándar tendrán un impacto beneficioso significativo en la productividad del usuario y del desarrollador.

REQUERIMIENTOS GENERALES PARA UN INTERFACE DE USUARIO GRÁFICO.

La aceptación de la industria, soportada por los vendedores de hardware y software y por las preferencias de los usuarios finales de una organización por una particular GUI, hace a la GUI un candidato potencial para un estándar GUI. Sin embargo, los estándares de interface de usuario gráfico para sistemas abiertos deberían satisfacer los siguientes requerimientos :

PORTABILIDAD :

Las aplicaciones y habilidades de los usuarios deberían ser transportables, a través de varias plataformas de sistemas abiertos. Los estándares GUI deberían de mantener un API estable para cada plataforma, y permitir así, un puerto fácil y rápido de una plataforma a otra. Un interface de usuario gráfica debería mantener una imagen y sentido consistente en todas las plataformas, y reducir entonces la necesidad de contraerse y facilitar la transferencia de habilidades requeridas del usuario.

FLEXIBILIDAD DE LOS ESTÁNDARES :

Este es el requerimiento clave del sistema abierto. Es asimismo una condición necesaria para un interface de usuario gráfica, para atraer grandes clientes corporativos y gubernamentales. Un estándar de facto mejor conocido para un interface de usuario gráfico de un sistema abierto, es el sistema Window MIT X. El sistema Window MIT X sirve de modelo para los estándares del Instituto Nacional de Ciencia y Tecnología (NITS), ANSI, e IEEE, y las especificaciones X/Open. El consorcio MIT X ha publicado un conjunto de especificaciones, que permiten a las aplicaciones del cliente comunicarse y trabajar conjuntamente. Estas especificaciones están enumeradas en el manual de convenciones y comunicaciones del intercliente (ICCCM) y son importantes para permitir un alto grado de interoperabilidad de las aplicaciones.

HERRAMIENTAS DE DESARROLLO :

Cualquier interface de usuario gráfica que está considerada como estándar, debe estar acompañada por un conjunto comprensivo de herramientas de desarrollo. Estas herramientas pueden agilizar el desarrollo de la aplicación, y permitir extensiones definidas para el desarrollador, y construir aplicaciones de GUI para una amplia variedad de plataformas.

FLEXIBILIDAD :

Un GUI estándar debe ser flexible y suficientemente extensible, para acomodar nuevos tipos de despliegue y otros dispositivos Entrada/Salida que estarán disponibles en el futuro.

INTERNACIONALIZACIÓN :

En los mercados globales de hoy, la internacionalización es otra manera de alcanzar la portabilidad en las aplicaciones. Esto incluye: otros idiomas, números, unidades monetarias, formatos de fecha y hora, y símbolos especiales y mensajes únicos para la cultura de ese país.

PLATAFORMA DE INDEPENDENCIA :

Para ser verdaderamente abiertos y estándar, un GUI debe ser diseñado para operar independientemente del sistema operativo o de la plataforma de hardware en la cual es ejecutado. De manera similar, en un ambiente de red un GUI estándar debe operar independientemente de los protocolos de la red en la que se fundamenta.

3.1.3 CARACTERÍSTICAS GUI

En general, la presentación de la información GUIs en áreas rectangulares en la pantalla, son llamadas ventanas (windows). Las ventanas pueden traslaparse una sobre otra. Los usuarios pueden ejecutar varias manipulaciones en las ventanas, como cambiar el tamaño y su posición. Las ventanas pueden contener objetos que pueden seleccionarse posicionando el mouse y presionando en los pequeños objetos llamados iconos. Una ventana completa puede ser minimizada en tamaño hasta llegar a ser un icono, y el usuario puede restaurar un icono a su tamaño normal (antes de ser minimizado).

Las interfaces de usuario gráfico avanzadas casi eliminan por completo la necesidad de escribir comandos, y permitir a los usuarios seleccionar comandos del menú, usando el mouse, o funciones de teclas.

Las ventanas pueden contener otras entidades gráficas (como barras de selección, botones, etc.), las cuales permiten a los usuarios controlar el contenido de las ventanas, y proveer entradas adicionales a la aplicación.

Cuando se compara con la programación tradicional, la diferencia más significativa en la lógica de presentación que controla la interface de usuario gráfica (GUI), es la noción que el usuario debe estar en control de la lógica. Por lo tanto, la programación estructurada tradicional con las secciones domésticas, sección de procesamiento y la sección de salida ha sido modificada. La programación GUI debería de ser capaz para aceptar procesos de eventos asincrónicos iniciados por un usuario o un sistema en cualquier momento.

TIPOS DE EVENTOS

El conjunto soportado de eventos de entrada generados por el usuario y generados por el sistema difieren de una implementación GUI a otra. Entre los de eventos comunes tenemos los siguientes :

- Eventos de Mouse, que ocurren cuando el usuario ha movido la posición del mouse, hacia adentro o afuera de una entidad, cuando presiona el botón del mouse, o presionando el botón del mouse adentro a afuera de la entidad o soltando el botón.
- Los eventos del teclado ocurren cuando el usuario ha presionado o liberado una tecla.
- Los eventos del menú ocurren cuando el usuario selecciona un comando del menú.
- Los eventos de actualización de ventana ocurren cuando una porción de la figura de una aplicación window ha sido dañada (una posible causa, es porque fue traslapada por otra ventana), y cuando ha sido redibujada.
- Los eventos de reacomodamiento ocurren cuando un usuario ha cambiado el tamaño de la ventana.
- Los eventos de activación/desactivación son generados por la interface de usuario gráfico (GUI), y permitir al usuario cambiar la ventana activa actual.
- Los eventos de inicio/fin ocurren cuando una entidad GUI ha sido creada o destruida; de este modo la aplicación puede necesitar una ejecución de una lógica de inicio y salida.

DISTRIBUCIÓN DE EVENTOS

Estos y otros eventos deben ser procesados por la lógica de presentación en forma cooperativa con la lógica de la aplicación. El procesamiento necesario es distribuido entre el GUI, la lógica de la aplicación y el interface de programación de la aplicación (API) de una particular GUI. Típicamente, un API es un conjunto de librerías de rutinas para una GUI específica, que ejecutan funciones como la creación de ventanas y despliegues de varias gráficas. Hay varios modelos para el procesamiento de eventos distribuidos: modelo de ciclo, modelo callback, modelo híbrido.

- El modelo de ciclo de evento especifica que una aplicación debe de contener un ciclo (loop) de eventos. El ciclo de eventos llama a una rutina de una librería en particular, para verificar si hay eventos pendientes. Cada evento pendiente obliga a la aplicación a despacharlo a una rutina manejadora de eventos, antes de regresar el control al ciclo de eventos. Para preservar la impresión del usuario que siempre está en control, la aplicación debe de retornar rápidamente al ciclo de eventos, aun si el procesamiento del evento no ha sido completado, (ver figura No. 3.1).

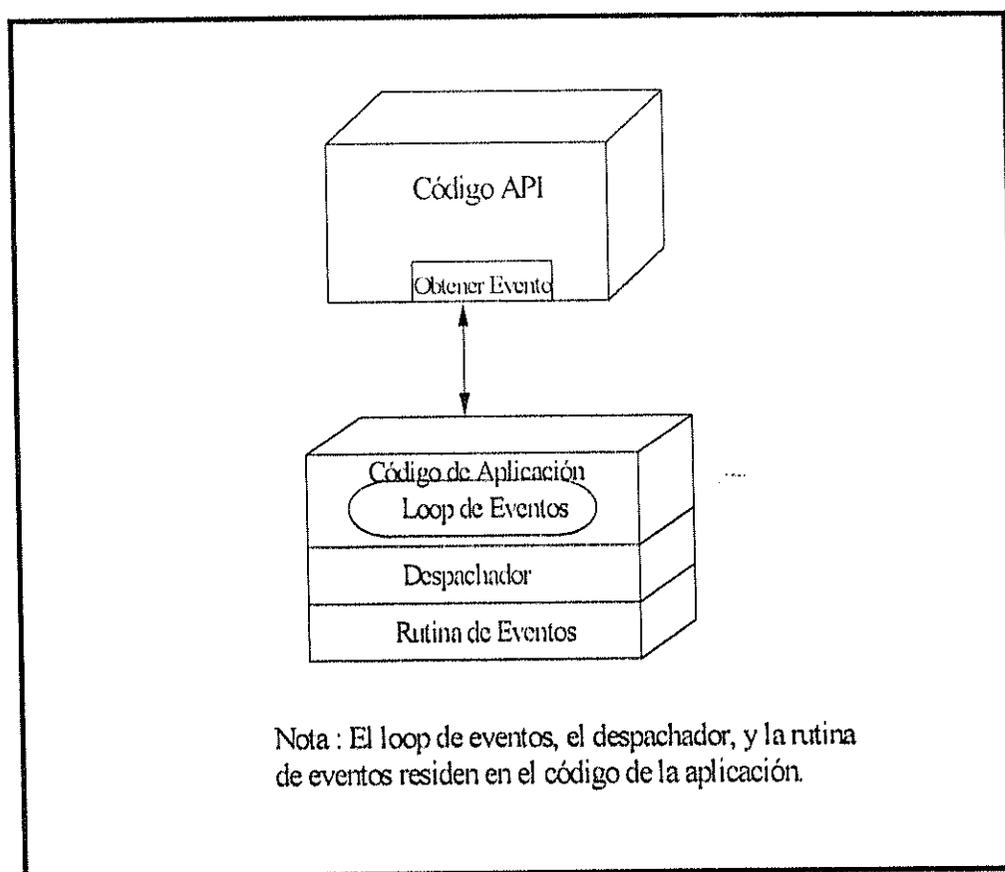


Fig. No. 3.1 Modelo de ciclo de evento.

- El modelo Callback de eventos requiere que la aplicación registre una función de manejo de eventos para cada entidad GUI que ésta crea; así libera a la aplicación de un sobre chequeo de ciclo de evento. Cuando la GUI detecta un evento (como el comando de un menú, o el teclado) de una entidad, llama a la rutina de eventos apropiada de la aplicación. La aplicación obtiene el control solamente en el tiempo de

la inicialización de la entidad y cuando la rutina de manejo de eventos es llamada por uno de los eventos, ver figura No. 3.2.

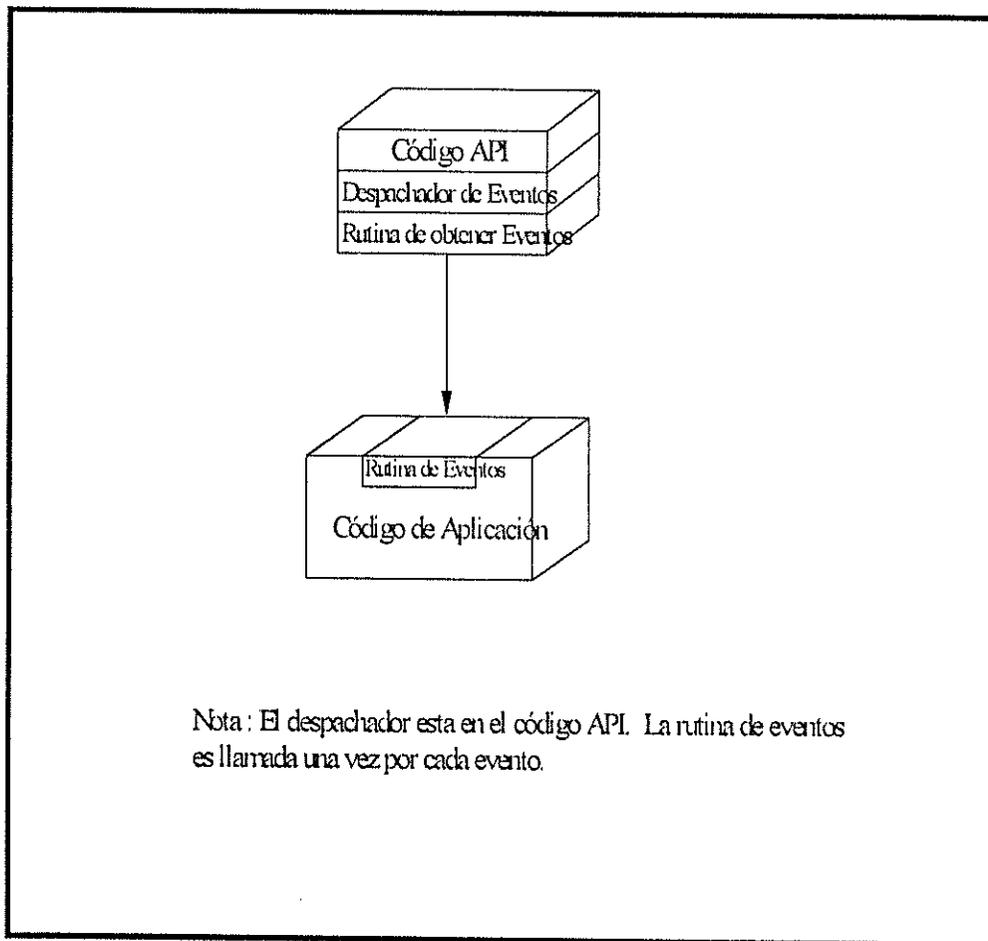


Fig. No. 3.2 Modelo CallBack de eventos.

- El modelo Híbrido combina el modelo de ciclo de eventos y modelo Callback de eventos.

CARACTERÍSTICAS DE SALIDA DE LOS GUI

Hay varias características de salida de los GUI que distinguen un GUI de otro.

Los espacios de coordenadas describen el sistema de coordenadas de dos dimensiones, que permite a los GUI direccionar píxeles individuales en la pantalla definida, en el punto de inicio e indicando la resolución de espacio de trazado.

La mayor parte de los GUIs el lugar de comienzo es la esquina superior izquierda, con las coordenadas que se incrementan hacia abajo y a la derecha. Los sistemas de coordenadas de diferentes GUIs pueden diferir en su resolución. La resolución es medida en puntos por pulgada (dots per inch o DPI), en el caso de terminales caracteres en un simple carácter (DPI no es aplicable en este caso). Las aplicaciones

deben estar enteradas del espaciado entre coordenadas y la resolución informándole al GUI en este orden, para pintar correctamente la figura en la pantalla.

Los algoritmos de dibujo describen la forma en particular de un GUI de dibujar líneas, centrar líneas y unir líneas. Los colores afectan la "imagen y sentido" de la presentación de los dibujos por el GUI. Desafortunadamente, los colores pueden diferir ampliamente de un dispositivo de despliegue con otro. Además, los GUIs por ellos mismos pueden tratar los colores de forma diferente y usar números diferentes de un bit para representar los colores, los cuales en un orden determinan la precisión de los colores que pueden ser seleccionados.

La presentación de texto en un ambiente gráfico es diferente de la presentación basada en carácter. El texto bajo el ambiente GUI es tratado como una gráfica, y puede ser desplegado con una amplia variedad de opciones. Esas opciones incluyen color, tamaño del carácter, el font y el estilo.

3.1.4 HERRAMIENTAS COMPATIBLES

La meta de las herramientas compatibles es proveer un común denominador para todos los GUIs. Con tales herramientas, los desarrolladores de aplicaciones llegarán a dominarlas y utilizar una herramienta para generar un interface de usuario gráfico deseado en una plataforma dada, en lugar de aprender varias interfaces diferentes, ver figura No. 3.3. Esto significa que, dependiendo de un GUI particular que se ha decidido usar en una plataforma, el código de la aplicación podría ejecutarse en un conjunto de funciones específicas de API de GUIs. Este conjunto varía de un GUI a otro.

Por ejemplo, cuando una aplicación que usa OSF/Motif GUI necesita ser portada a una plataforma MS-DOS soportando GUI Windows, la aplicación no debería ser cambiada. Todos las interacciones GUI en este caso estarían manejadas a través de una herramienta API común, las herramientas que hacen puente en las diferencias entre varias implementaciones GUI, deberían tomar en cuenta las diferencias entre los interfaces de programación de aplicación (API), lógica de manejo de eventos, espacios entre coordenadas, algoritmos de dibujos, colores, texto y fonts usados por varios GUIs.

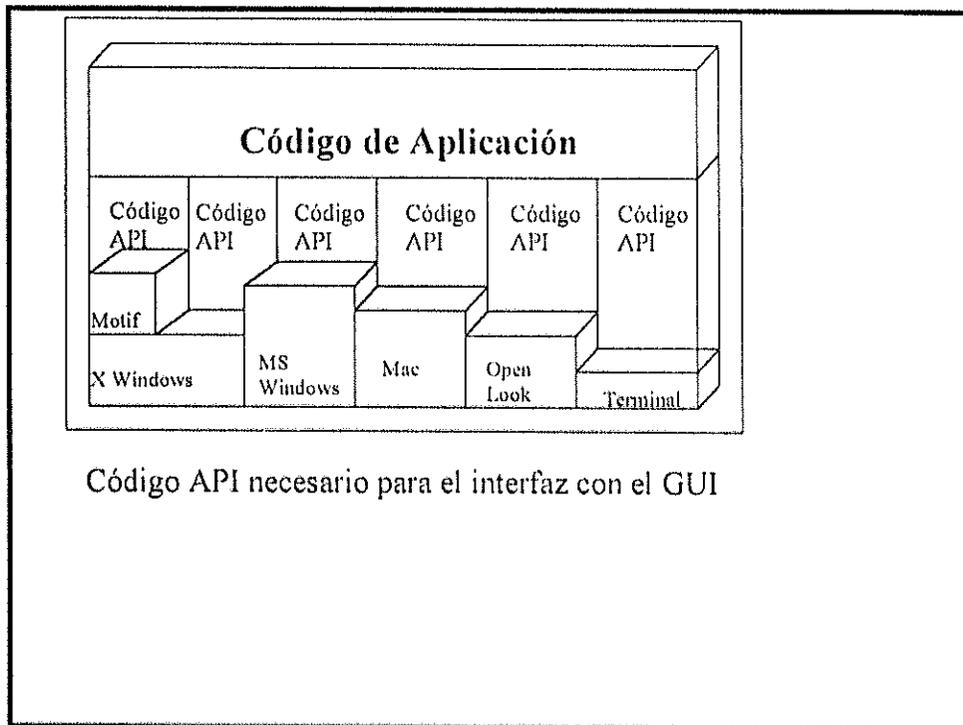


Fig. No. 3.3 Varios APIs GUI.

Al mismo tiempo, esas herramientas deberían retener en la mayor cantidad posible, la imagen y sentido del GUI. Otro requerimiento para tales herramientas es rechazar el acercamiento de un mínimo común denominador, el cual simplemente elimina las ventajas de un GUI particular; en vez de ello, tales herramientas deberían proporcionar una serie de funciones que fueran al menos tan completas como las más avanzadas de los GUIs soportados y preferiblemente, aun más.

Dada la amplia variedad de niveles de soporte nato de interface GUI, la creación un API para este tipo de herramienta no es una tarea fácil. Existen otros requerimientos que deberían estar soportados por tal herramienta de compatibilidad :

- Las aplicaciones escritas usando la herramienta, deberían ejecutarse al menos correctamente, como si hubiera sido escrita en un API de GUI nato (inicial).
- La herramienta o un subconjunto razonable de ésta debería estar disponible para caracteres o terminales.
- Las herramientas deberían ser flexibles y extensibles en este orden para soportar futuros GUIs, nuevas entrada/salidas y dispositivos de despliegue y multimedia.

Dada la necesidad y las ventajas de tal herramienta, no es sorpresa que varios vendedores empiezan a desarrollar productos compatibles. Uno de ellos es la Corporación Oracle, que ésta trabajando en la versión de herramienta de compatibilidad, Oracle toolkit, actualmente Oracle toolkit es diseñada para proporcionar un API común para las siguiente GUIs : Macintosh, Microsoft Windows, X Windows, OSF/Motif y terminales caracter.

La compatibilidad de herramientas como Oracle toolkit, permitirán a los desarrolladores mantener en paso con los avances en la tecnología GUI, y facilitar la portabilidad y consistencia de la presentación de las aplicaciones, mientras que los estándares GUIs están empezando a desarrollarse.

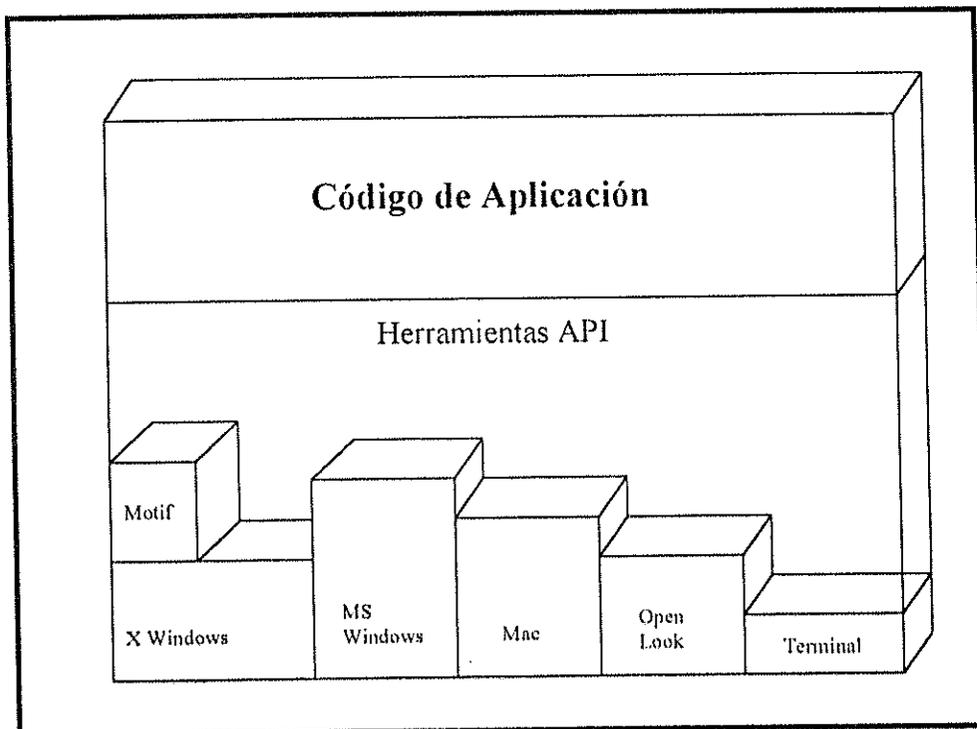


Fig. No. 3.4 Compatibilidad de Herramientas APIs.

3.2 SERVIDORES

Una de las principales claves detrás del modelo de computación cliente/servidor, es la separación física de la administración de la presentación de los usuarios, de otros servicios de la aplicación. En el complejo ambiente distribuido cliente/servidor, la idea es ayudar a la evolución de los sistemas cliente/servidor hacia la especialización, que es dirigida a tareas funcionales particulares en un solo camino. Esta especialización ha empezado a mostrar las facilidades de las funciones del cliente, como las interacciones de usuario-cliente, en énfasis en la funcionalidad de la presentación.

Los nodos de servidores pueden llegar a especializarse para ejecutar las funciones del servidor con una mayor eficiencia. Este capítulo examina las funciones típicas del servidor; ciertas características de diseño especializado usadas para facilitar esas funciones, algunos ejemplos de implementaciones de servidores especializados.

3.2.1 FUNCIONES

La especialización del servidor es el mejor reflejo en la funcionalidad y diseño de servidores de bases de datos. Básicamente, los servidores de bases de datos deberían estar disponibles para proveer grandes cantidades de almacenamiento en disco rápido, poder de procesamiento significativo, y la habilidad para correr muchas aplicaciones (clientes) simultáneamente. Sin embargo, la tecnología continua desarrollándose, y la especialización es extendida a funciones de comunicación, emulación de terminales, fax, administración de librerías, y correo electrónico (E-mail).

Lógicamente, un servidor es un proceso lógico que provee servicios a los procesos requerientes. En la computación cliente/servidor, un cliente inicializa la interacción cliente/servidor enviando un requerimiento al servidor. Las funciones que el servidor debería de ejecutar son determinadas, en gran parte, por los tipos de requerimiento que los clientes pueden enviar a los servidores. Recíprocamente, si un servidor está imposibilitado para ejecutar una función requerida por el cliente, este servidor no puede participar en las interacciones cooperativas cliente/servidor. Idealmente, un cliente no debería de enviar un requerimiento al servidor, si éste no soporta.

En general, sin embargo, una vez los clientes y servidores están interconectados en una red, las siguientes funciones pueden ser requeridas a los servidores por los usuarios (ver figura No 3.5) :

- **Compartir archivos.** En ambiente de trabajo de grupo, los clientes pueden necesitar compartir la misma información del archivo. Una amplio porcentaje de los archivos están localizados en un servidor de archivos, con procesamiento de archivos compartidos, y los clientes envían sus requerimientos de entrada/salida de archivo al servidor de archivos. Generalmente, un servidor de archivos provee acceso a un cliente a un archivo completo, cuando éste actualiza el archivo ningún otro cliente puede acceder este archivo. Otro uso típico de los servidores de archivos es la transferencia de archivos entre clientes.
- **Compartir impresoras.** En un ambiente de trabajo en grupo, una impresora con gran capacidad puede reemplazar todas las impresoras individuales de los clientes. Luego todos los clientes pueden enviar el requerimiento de impresión de un archivo al servidor de impresora. Un servidor de impresión mantiene una cola de todos los archivos que serán impresos, y enviar cada archivo de impresión por su turno a la impresora compartida (usualmente de impresión rápida, de alta calidad de impresión). Típicamente,

todos los archivos de impresión individualmente son impresos con separadores de páginas indicando el nombre del cliente y el archivo.

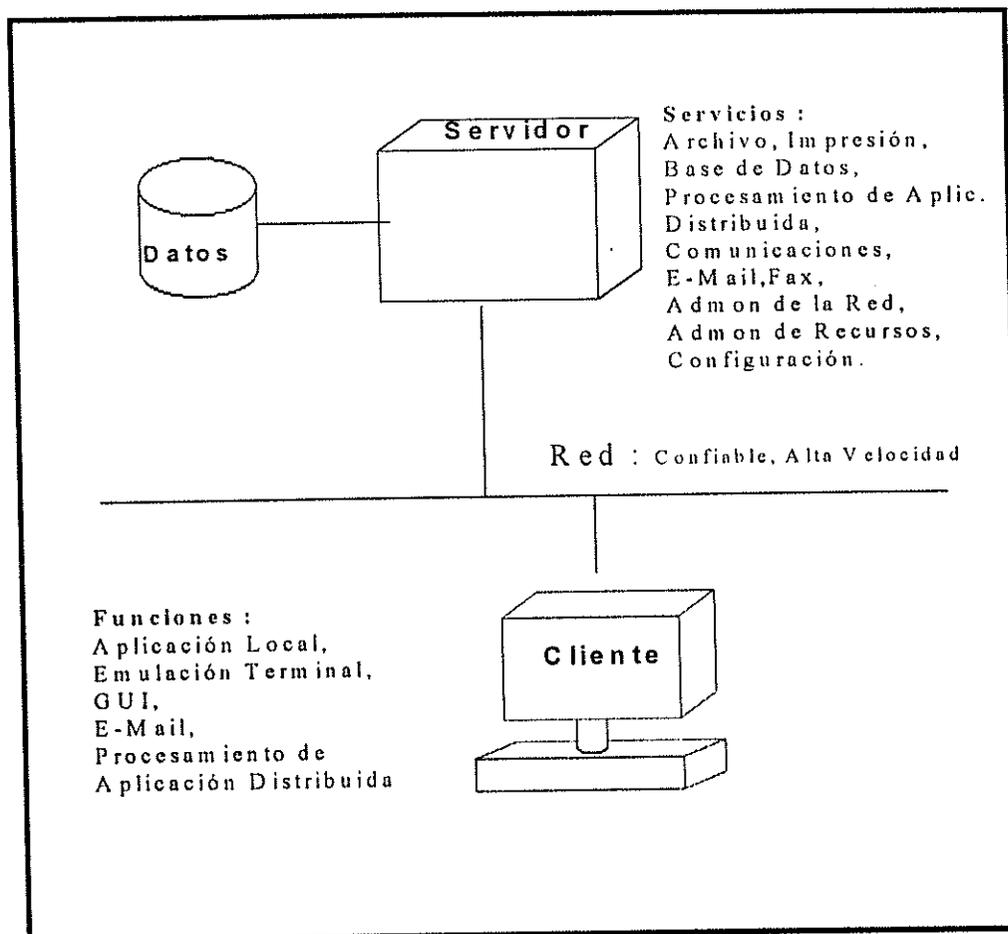


Fig. No 3.5 Funciones Cliente/Servidor

- **Acceso a la base de datos.** En el ambiente cliente/servidor, el procesamiento de la aplicación es dividido entre el sistema del cliente y el sistema del servidor. Los servidores pueden ejecutar alguna porción de la lógica del negocio y la lógica de la base de datos. De manera similar a los servidores de archivos, los servidores de bases de datos proveen acceso a los clientes a la información que reside en el servidor. Sin embargo, el sistema administrador de la base de datos (DBMS) es más sofisticado que los métodos de acceso básico E/S de archivos. El DBMSs provee acceso a los datos concurrentemente con varios niveles de bloqueo granular e integridad de los datos. El DBMS elimina la información redundante, y permiten una distribución de la información transparente a los usuarios, y permite que partes de la lógica de acceso de información de una aplicación específica sea incorporada por el mismo DBMS. Los clientes solicitan acceso a la información deseada (contrario al acceso del servidor de archivos al archivo completo), y toda la manipulación necesaria de la información requerida es desarrollada en el servidor de base de datos. De esta manera, múltiples clientes pueden acceder a la base de datos concurrentemente.
- **Servicios de comunicación.** En un ambiente de trabajo de grupo que está conectado a un procesador Host remoto, todo el hardware y software de comunicación puede ser concentrado en un servidor de

comunicación especial, en los cuales los clientes pueden dirigir sus requerimientos de comunicación para ser procesados.

- Servicios de Facsímil, los cuales usualmente requieren software y equipo especial, son ahora más frecuentemente asignados a servidores de fax dedicados.

Otras funciones requeridas por el cliente, tales como el correo electrónico, librerías, redes de trabajo, administración y configuración del recurso, están siendo ahora manejadas en el ambiente actual cliente/servidor a través de servidores apropiados.

Un nodo servidor, en un modelo cliente/servidor, puede especializarse para desarrollar su función particular de la manera más eficiente. Sin embargo, además de la especialización de funciones específicas individuales, los servidores como una clase de sistemas, pueden especializarse para satisfacer los siguientes requerimientos para propósitos generales :

- Soporte MultiUsuario. Aun en un pequeño ambiente de trabajo de grupo, un servidor debería de ser capaz de dar servicio a múltiples clientes concurrentes. Los clientes, que están corriendo diferentes tareas, esperarían que un servidor soportara el procesamiento multitarea. Obsérvese que las multitareas pueden implementarse en un sistema de usuario individual (como OS/2), y es un requerimiento necesario pero insuficiente para soporte multiusuario (un sistema multitarea no es igual a un sistema multiusuario). Los requerimiento de soporte multiusuario asimismo incluyen características importantes como la protección de memoria y multinivel.
- Escalabilidad. De la misma manera que el número de aplicaciones, sus requerimientos de recurso, y el número de usuarios se incrementa, un servidor debería ser capaz de satisfacer estas crecientes demandas, con base en sus propios recursos (es decir, debería de proporcionar desempeño escalable). La escalabilidad no significa que los usuarios deban adquirir un servidor de sistemas con demasiada capacidad adicional, aun con costo extra. Por el contrario, el sistema debería satisfacer los requerimientos actuales y, al mismo tiempo, debería tener facilidad de expansión (sería posible actualizando el CPU, o los discos duros). Una opción menos atractiva de la escalabilidad sería reemplazar el sistema cada vez que alcanza el límite de su capacidad.
- Desempeño. Un servidor de sistema debería proporcionar niveles de desempeño satisfactorios para las necesidades del negocio y requerimientos del usuario en un ambiente cliente/servidor multiusuario. Aun cuando los requerimientos del negocio no implican un tiempo de milisegundos, cada vez que hay una transacción de negocios, los usuarios difícilmente apreciarían un sistema que toma más de unos pocos segundos en responder a cada acción del usuario.
- Almacenamiento. Conforme el número de usuarios y aplicaciones corriendo en un servidor se incrementan, y los avances en tecnología de almacenamiento conllevan a la baja de los costos de almacenamiento físico, la demanda para almacenamiento extra y acceso más rápido, se convierte en uno de los requerimientos críticos para un servidor de sistemas. Las demandas de almacenamiento provienen de los sistemas operativos que necesitan almacenamiento adicional, para implementar nuevas características de usuarios que desean guardar varios archivos de información en un servidor, y de aquellas aplicaciones tales como las herramientas DBMSs y CASE, que son algunos de los mayores consumidores de almacenamiento. Por ejemplo, si una estación de trabajo corriendo una herramienta CASE necesita al menos 16 Mbytes de memoria RAM y 300 Mbytes en disco duro, un servidor podría

requerir de 32 a 62 Mbytes de memoria, y de 1 a 1.5 Gbytes de disco duro para soportar varias de estas redes de trabajo.

- **Multimedia.** Conforme las nuevas aplicaciones y tecnologías están disponibles, la demanda para soporte de almacenamiento de multimedia se incrementa. Aplicaciones de imagen, video y sonido, son cada vez más populares. Es así como los requerimientos para un servidor de sistema pueden incluir la habilidad para guardar no solamente imágenes digitales en discos duros, si no que también hipertextos en un dispositivo de almacenamiento óptico de una sola escritura y varias lecturas (Write-Once-Read-Many WORM), y datos de sonido y en video cassettes, discos compactos y video discos.
- **Redes de Trabajo.** Las comunicaciones cliente/servidor se realizan sobre una red de trabajo de comunicación. Los sistemas, tanto del cliente como del servidor, deberían tener capacidades de redes de trabajo construidas. Sin redes de trabajo no hay interacción cliente/servidor y por lo tanto no hay clientes ni servidores. Si un sistema está diseñado teniendo en mente los requerimientos de las redes de trabajo, las arquitecturas de software y hardware de sistemas, pueden ser integradas de una manera óptima con las interfaces de redes de trabajo y protocolos.

Una conclusión, que puede ser tomada del análisis de estos requerimientos, es que los sistemas mainframe parecen ser los mejores candidatos para servidor de plataforma. En lo concerniente a capacidad y funcionalidad, los mainframes ciertamente están bien situados, para ser servidores en un computador cliente/servidor; de hecho, la dirección establecida de los proveedores de computadores como los host mainframes de IBM, juegan un papel de servidor en un ambiente cliente/servidor, que requiere y obtiene acceso para grandes cantidades de información corporativa almacenada en varios host DBMS (DB2, IMS, IDMS, etc.). De hecho los sistemas de procesamiento transaccionales basados en mainframes, como los sistemas de información de control de clientes (CISC) de IBM, hoy día soportan miles de usuarios corriendo en aplicaciones en línea, que simultáneamente accesan la información en DB2, IMS, IDMS y archivos VSAM. Las aplicaciones mainframe pueden ser distribuidas de tal manera que el procesamiento de la aplicación sea ejecutado cooperativamente en la arquitectura cliente/servidor. Tales implementaciones existen hoy día, por ejemplo, las aplicaciones basadas en los sistemas de información de control de clientes corriendo en un host MVS, con la parte frontal de la lógica de presentación ejecutadas por herramientas corriendo en estaciones de trabajo gráficas, ver figura No. 3.6.

Sin embargo, el uso del host y servidores puede ser una desventaja, desde el punto de vista de escala económica. Nuevas estaciones de gran poder permiten a los usuarios finales tomar ventaja de un significativo beneficio precio/rendimiento de la tecnología de minicomputadores. Un típico mainframe cuesta muchísimo más que una estación de trabajo, con la misma velocidad de procesamiento.

El poder de procesamiento es frecuentemente medido en MIPS (millones de instrucciones por segundo), MLFLOPS (millones de operaciones de punto flotante por segundo), o más usado por los usuarios finales, el número de transacciones por segundo (TPS) de una aplicación dada que es procesada en una plataforma dada. Si el volumen de la transacción y los todos los requerimientos son conocidos, el costo de las transacciones por segundo puede ayudar a determinar cuanto hay que gastar para soportar una aplicación de negocios en particular.

La economía de escala llega a ser importante cuando los usuarios comparan las características típicas de precio/rendimiento, en sistemas de gran escala (aproximadamente \$ 100,000- \$200,000 en ambientes DB2 MVS IBM medido en TPS) con el costo de un sistema con una capacidad similar, construido sobre RS/6000-520 de IBM o un servidor Sparc de SUN (aproximadamente \$1,000 - \$4,000, en TPS). La misma

relación puede ser observada en los costos por MIPS comparado entre un mainframe y un servidor con un promedio de velocidad de 25 MIPS⁷.

Las ventajas del precio/rendimiento de los servidores, basados en microcomputadoras, es una de las razones de que la arquitectura cliente/servidor haya empezado a tener aceptación por un número creciente de empresarios. El poder y la capacidad de los mainframes asegura su rol crítico como un servidor central de una corporación, y como la mejor plataforma central de repositorio de información.

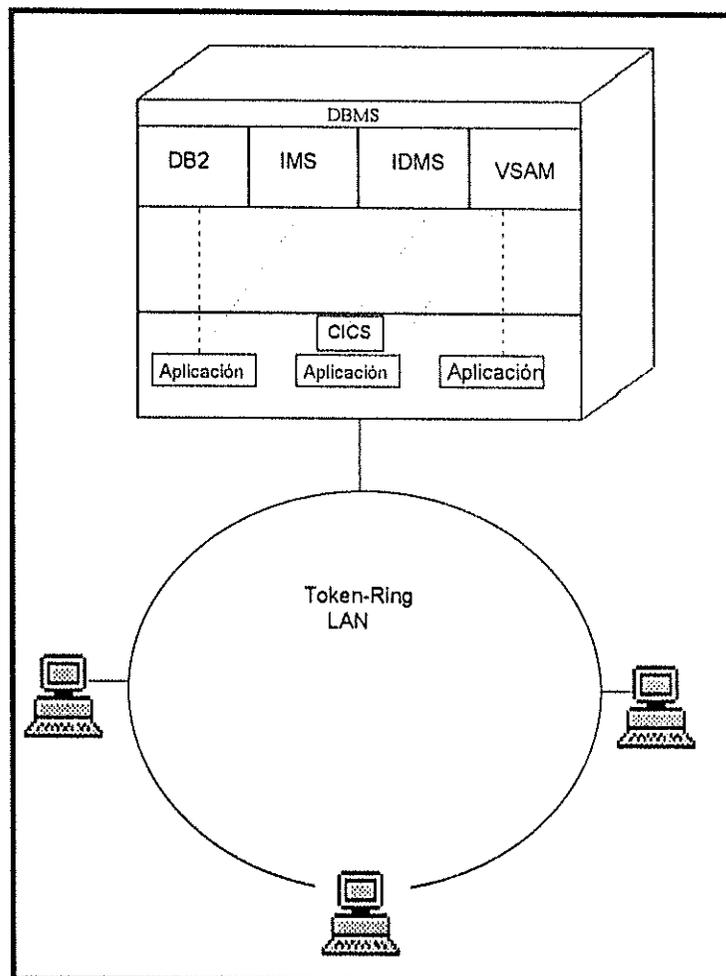


Fig. No. 3.6 Mainframe como Servidor.

Sin embargo, no todos los negocios necesitan un poder y la capacidad de almacenamiento de computación de un mainframe. Ni todos los negocios y la aplicaciones más importantes justifican la millonaria inversión de un mainframe y el costo elevado de mantener tener la información centralizada.

⁷ Tomado del libro Client/Server Architecture, Alex Berson, Pag 82, 1992.

3.2.2 ARQUITECTURA HARDWARE DEL SERVIDOR

La discusión de los aspectos de gran rendimiento de la especialización de la plataforma del servidor, llegaran a concentrarse en la especialización y avances de la arquitectura hardware del servidor. Este análisis de la tendencia de la arquitectura de hardware del servidor, está destinada a clarificar el mal uso y la mala interpretación que se le da frecuentemente a la jerga tecnológica que emplean algunos vendedores de hardware, y al mismo tiempo, dar a los lectores un punto de referencia que puede ser usado cuando se tome la decisión de comprar un servidor. El punto de la discusión está centrado en dos arquitecturas de hardware ampliamente populares, reducidas a las características de las arquitecturas RISC (Reduce Instruction Set Computer, ver figura No. 3.7) y SMP (Symetric MultiProcessing).

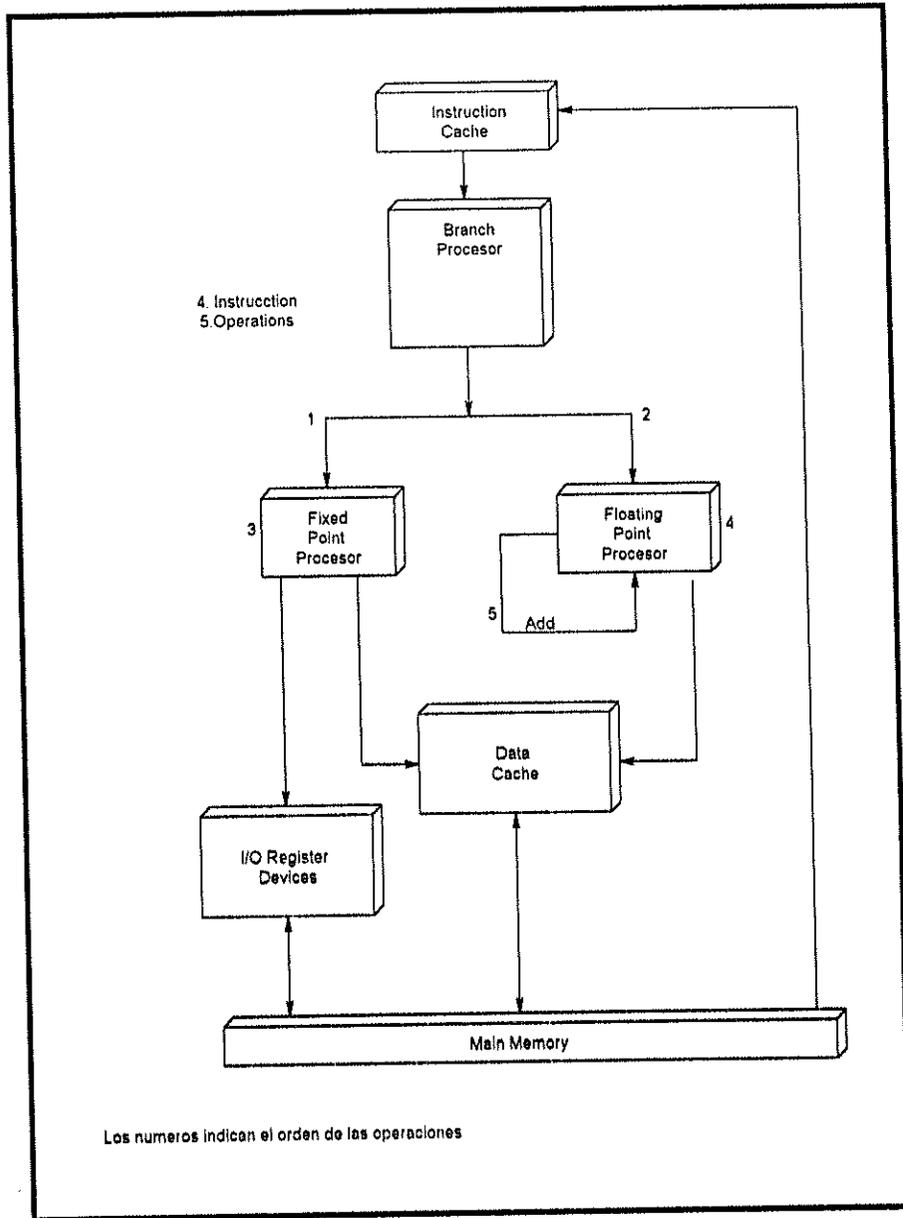


Fig. No. 3.7 Arquitectura RISC RS/6000 Superescalar.

CONSIDERACIONES DE SISTEMA.

Los principios básicos del diseño de computadoras describen, entre otros aspectos, los factores que afectan al rendimiento de las computadoras. En general, esos factores incluyen: la arquitectura CPU, el tamaño y los métodos de implementación del conjunto de instrucciones, la habilidad de los compiladores para optimizar el rendimiento, la tecnología de chip de la computadora, y el sistema operativo. Desde el punto de vista de las aplicaciones, la arquitectura CPU, la tecnología y el conjunto de instrucciones son menos importantes que los compiladores y los sistemas operativos, como las aplicaciones hoy día son escritas bastante más en lenguajes de alto nivel como C que assembler.

Los compiladores y los sistemas operativos, por consiguiente, deberían de ser capaces de explotar la arquitectura del hardware, para llegar al más alto rendimiento posible. Así el conjunto de instrucciones por debajo llegará a ser uno de los factores de rendimiento crítico. De hecho, el rendimiento de la computadora puede ser descrito por la siguiente fórmula simbólica :

$$\text{Performance} = 1/[(\text{Tiempo de ciclo}) \times (\text{Longitud del camino}) \times (\text{Ciclo/instrucción})]$$

El tiempo de ciclo es contrario a la velocidad de reloj del sistema y está principalmente limitado por la base de la tecnología del chip. Hoy día, la mayor parte del equipo de computación y las estaciones de trabajo pueden operar a velocidades de reloj de 12 a 33 MHz, especialmente sobre los 50 MHz (50 millones de ciclos de reloj por segundo), pueden requerir nuevas tecnología de chip en alta densidad. El acortamiento del ciclo de tiempo eleva el rendimiento.

La longitud del camino describe el número de instrucciones de máquina necesarias para ejecutar un comando. El acortar la longitud del camino eleva el resultado del rendimiento. La arquitectura CPU, con base de un conjunto de instrucciones y la optimización de los compiladores, son factores que pueden reducir la longitud del camino.

Ciclo/Instrucción; este factor describe cuántos ciclos de computadora son necesarios para ejecutar una instrucción. Este número puede variar, de menor que uno (en reducir en arquitectura de computadoras RISC) a mayor que uno, en las arquitecturas tradicionales. Si el número es menor que uno, esto implica que más de una instrucción puede ser ejecutada en el ciclo de CPU, así mejorar el rendimiento.

Otros factores que afectan el rendimiento de la computadora, incluyen; el tiempo de acceso a la memoria, las características de acceso externo y la velocidad de E/S al transferir información, y el ambiente de la red, comunicaciones y redes.

Algunos sistemas pueden ser bien optimizados, para los ambientes comerciales, los cuales son generalmente caracterizados por el procesamiento entero, proceso transaccional, la manipulación de los subsistemas de archivo y disco y un número significativo de terminales con funciones media-baja y estaciones de trabajo conectadas. Otros sistemas pueden mejorarse para satisfacer los ambientes científicos, los cuales generalmente se caracterizan por un alto requerimiento de ejecución de punto flotante, y pocas funciones gráficas en las estaciones de trabajo conectadas al servidor central, vía interconexiones de alta velocidad. Todavía otros sistemas pueden ser desarrollados para alcanzar un cuidadoso balance entre el rendimiento entero y de punto flotante, y así extender la aplicabilidad del sistema al mundo científico y comercial.

En cualquier evento, cuando los diseñadores de sistemas desean dirigirse al emisor del rendimiento, ellos pueden concentrar sus esfuerzos en uno de los siguientes puntos:

- Acortando la longitud del camino de la instrucción.
- Mejorando la proporción de instrucciones por ciclo,
- Acelerando el reloj del sistema.

La ultima opción puede ser aplicada al procesamiento entero o de punto flotante, y limitar así la aplicabilidad del sistema a un particular ambiente. Recíprocamente, los diseñadores pueden intentar alcanzar el rendimiento deseado para mezclar instrucciones enteras y de punto flotante, para crear así una arquitectura universal de alto rendimiento.

3.2.3 MULTIPROCESAMIENTO SIMÉTRICO.

El multiprocesamiento es una herramienta conveniente e indispensable para mejorar el rendimiento de los sistemas de la computadora, y la hace soportar continuamente demandas de aplicaciones más complejas. Como los costos de CPU están decreciendo, los usuarios descubren que agregando procesadores a su hardware existente en el multiprocesador, es significativamente más económico, comparándolo con agregar una nueva computadora al sistema o reemplazando el sistema existente con un sistema uniprocador más poderoso. El agregar o reemplazar uniprocadores tiene sus desventajas :

- Agregar procesadores usualmente implica agregar dispositivos periféricos caros.
- Un uniprocador solo no puede compartir memoria, a menos que ellos sean conectados a la red con costos adicionales, ni tampoco pueden elevar la velocidad de la aplicación, a menos que la aplicación esté distribuida con adicional costo y complejidad.
- Agregar un uniprocador rápido es más caro, que obteniendo el mismo rendimiento por el incremento del número de procesadores.

Así, el sistema basado en multiprocesamiento provee de habilidad para incrementar el rendimiento gradualmente. La escalabilidad resultante permite actualizaciones futuras, y representa soluciones de costo efectivas para las aplicaciones DBMS, pasarelas (gateways), y procesamiento transaccional. Típicamente, esas aplicaciones son compuestas de tareas pequeñas, relativamente independientes que pueden ser asignadas a numerosos procesadores. Aplicaciones como éstas son ideales para que corran en servidores en un ambiente cliente/servidor, por consiguiente, los servidores basados en multiprocesadores representan otro ejemplo de la especialización del servidor.

Hay dos tipos genéricos de sistemas de multiprocesamiento :

- Multiprocesadores conectados libremente o de memoria distribuida, incorporan un gran número de procesadores autosuficientes, cada uno con su propio sistema operativo y recursos (memoria, discos, etc.). El procesador se comunica a través de la red y los canales de E/S, los cuales hacen la

comunicación y cooperación interprocesadores dependiente del desempeño de la red de trabajo y del canal entrada/salida, por lo tanto, relativamente lento.

- Multiprocesadores conectados conjuntamente o de memoria compartida, consisten en un número relativamente pequeño de procesadores que comparten memoria en común, un sistema operativo común, E/S comunes, y otros recursos del sistema. Los procesadores seleccionan tareas para ser ejecutadas de un concentrador de tareas comunes, y son interconectados vía un bus del sistema de alta velocidad, y proporcionan de este modo una comunicación interprocesadores extremadamente eficiente.

De estos dos tipos genéricos de multiprocesamiento, solamente el multiprocesamiento de memoria compartida proporciona realmente rendimiento escalable a un bajo costo.

ARQUITECTURA DE HARDWARE ESCALABLE

El multiprocesamiento de memoria compartida asegura que cualquier procesador que está completando una tarea es puesto inmediatamente a trabajar en otra, que es la siguiente tarea disponible. El multiprocesamiento de memoria compartida típicamente incorpora un número de procesadores, llamados elementos de procesamiento (processing elements o PEs), que comparten memoria común, E/S común, y otros recursos comunes del sistema. Todos los PEs ejecutan una copia sencilla de un sistema operativo, y cada PE ejecuta una tarea seleccionada del concentrador de tareas comunes. Típicamente, el orden para reducir el volumen de tráfico de memoria compartida, cada PE tiene una o más memoria cache. Uno o más controladores de memoria pueden ser incluidos en esta arquitectura para soportar requerimiento de acceso de memoria grandes de múltiples PEs. Los PEs y los controladores de memoria están interconectados vía el bus del sistema de alta velocidad, y proporcionan comunicaciones interprocesadores de alta eficiencia.

Cuando todos los PE en un multiprocesador de memoria compartida tienen las mismas capacidades, y pueden ejecutar las mismas funciones, esto es llamado multiprocesamiento simétrico (SMP o Symmetric Multiprocessor), ver figura No. 3.8. Cada PE en un sistema SMP puede correr aplicaciones de usuarios correctamente, como cualquier porción del sistema operativo, incluyendo las operaciones E/S de interrupts, funciones del kernel del sistema operativo, e E/S de los drivers. Adicionalmente, cualquier tarea puede ser ejecutada en cualquier PE y puede migrar de un PE a otro PE.

Las implementaciones SMP tienen un significativo efecto positivo en la escalabilidad, de hecho, si un sistema designa un PE para ejecutar una tarea particular de servicio de interrupción E/S, por ejemplo (en una arquitectura no SMP) el PE llegará a sobrecargarse por el incremento en el número de requerimientos E/S, y la sobrecarga llegará a degradar el rendimiento del sistema.

La arquitectura SMP proporciona dos características de alto nivel :

- Ejecución flexible⁸ es la habilidad de un sistema SMP de soportar aplicaciones existentes, flexible y transparentemente al usuario, realmente en las implementaciones libres o flexibles, todas sus aplicaciones fueron escritas originalmente para uniprocadores y deberían de ser capaces de correr sobre un sistema SMP sin modificarlas. Tomando las ventajas completas del multiprocesamiento, las aplicaciones pueden ganar un significativo rendimiento. Esto es importante para servidores corriendo DBMSs y soportando procesamiento transaccional.

⁸ traducción de seamless : flexible, libremente, sin ataduras.

- El rendimiento escalable coordina dos componentes : el crecimiento computacional y el crecimiento E/S. El crecimiento computacional puede ser alcanzado agregando más elementos de procesamiento, mientras que el crecimiento E/S puede ser el resultado agregar más periféricos y/o buses E/S. Es importante notar que mientras la adición de un PE incrementa el rendimiento del sistema en general, el incremento no es igual al rendimiento de un PE individual, pero en lugar de esto, el rendimiento corresponde a una fracción del rendimiento PE. Factores tales como: el bus del sistema y diseño de subsistemas de memoria y porciones de memoria cache, afectan incrementando el rendimiento resultante.

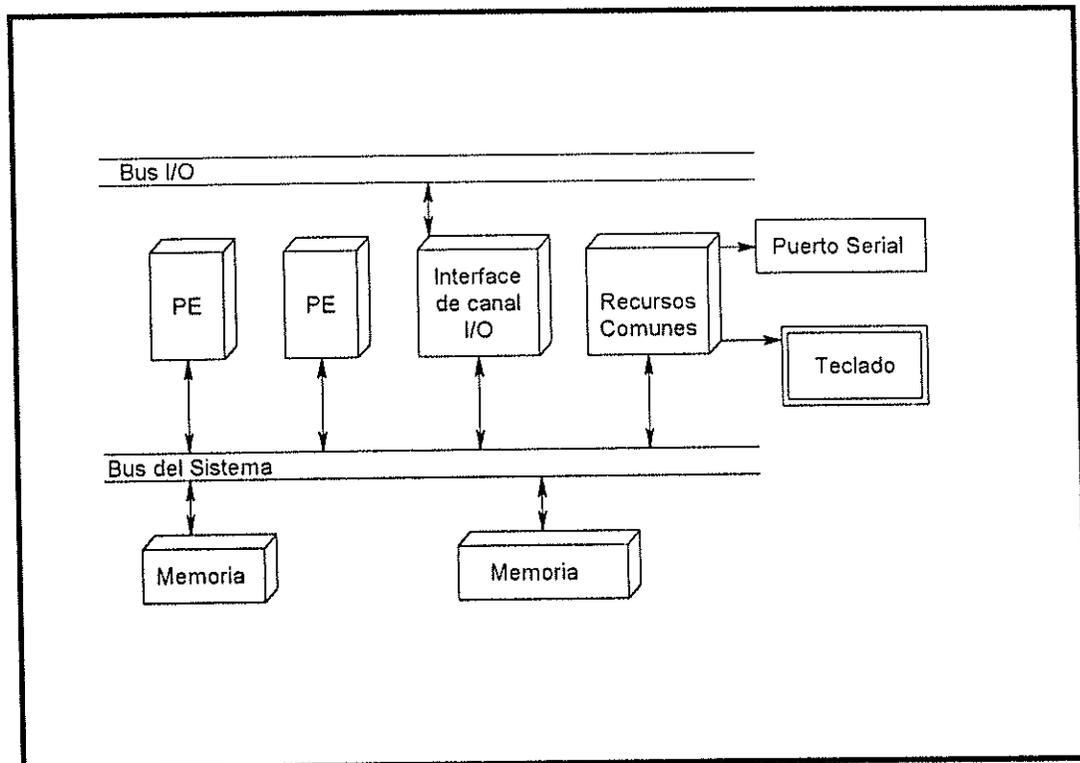


Fig. No. 3.8 Multiprocesador de memoria compartida simétrico.

ARQUITECTURA DE SOFTWARE ESCALABLE

Proporcionar simetría en los sistemas de memoria compartida, ha impactado a las arquitecturas de software y hardware. Las características de software que soportan rendimiento escalable SMP, incluyen capacidades para ejecutar operaciones del kernel del sistema operativo, E/S de interrupciones y E/S de los drivers en todos los PEs.

El diseño del sistema operativo llega a ser más complejo en un sistema SMP.

La estrategia de seguridad de los sistemas operativos está diseñada para prevenir actualizaciones simultáneas, a las estructuras de datos y códigos en la memoria compartida en el sistema SMP. La estrategia de seguridad representa un importante requerimiento para la arquitectura de software escalable.

El balance de carga dinámico es un requerimiento crítico en la arquitectura de software SMP. De hecho, si un PE ha sido agregado a un sistema SMP, el sistema operativo y la aplicación de software deberían ser capaces de tomar ventaja de la adición del PE por la redistribución dinámica de una cantidad de carga en todos los PE disponibles.

Las arquitecturas de software escalables están diseñadas no solamente para tomar ventaja de la arquitectura del hardware SMP transparentemente y flexible, ellas también proporcionan un fácil acceso a la computación paralela, donde múltiples aplicaciones pueden correr simultáneamente. Hay muchas técnicas para el desarrollo de aplicaciones paralelas capaces de lograr el máximo de rendimiento en un sistema SMP.

Entre ellos hay problemas divisionales (dividir un problema entre partes pequeñas independientes) y descomposición de la información para una carga balanceada.

CAPITULO 4 REDES Y COMUNICACIONES

4.1 SISTEMAS DE COMUNICACIÓN

Los sistemas cliente/servidor no están restringidos a redes de área local; pueden estar interconectados a través de cualquier tipo de red.

En los ambientes de comunicación, varios recursos del sistema (datos, poder de computación, programas etc.), están distribuidos a través de varias localidades. Esos recursos utilizan alguna clase de sistema de comunicación para interactuar con otro sistema. En este contexto, un sistema de comunicación es un mecanismo que permite distribuir recursos para intercambiar el control de la información y datos. Los sistemas de comunicación, mientras sean la esencia de la distribución, pueden ser implementados en un camino totalmente transparente para el usuario final. Consecuentemente, éstos pueden ser suficientemente visibles para el usuario final en cualquier implementación. La comunicación entre nodos es necesaria, y requiere alguna clase de red de trabajo física que conecte a todos los nodos que interactúan.

4.1.1 COMUNICACIONES Y DISTRIBUCIÓN

La arquitectura cliente/servidor es un caso especial del procesamiento cooperativo. El procesamiento cooperativo, en su orden, es un caso especial del procesamiento distribuido. Un ambiente de procesamiento distribuido no está motivado solamente por las necesidades del modelo computacional cliente/servidor. Entre los factores que contribuyen al interés actual en los sistemas distribuidos se encuentran :

- Los avances tecnológicos en microelectrónica, que han cambiado el porcentaje precio/rendimiento en favor del bajo costo/sistemas de alto rendimiento.
- El costo de las intercomunicaciones y comunicaciones han ido disminuyendo dramáticamente desde hace algunos años.
- Los usuarios demandan facilidades, más económicas, más rápidas, más sofisticado, y confiables.

Un objetivo de beneficio de la distribución es el de compartir los recursos. Una cantidad de recursos, como las computadoras, periféricos, procesadores, programas, datos, etc., son interconectados por un sistema de comunicación en un orden que permita compartir los recursos. La interconexión de los sistemas a una red, la cual es usada por un sistema de comunicación para intercambiar mensajes, o paquetes de información entre diferentes lugares, sistemas, terminales y programas.

Un sistema de comunicación es la colección de hardware y software, que soporta la comunicación intersistemas e interprocesos entre los componentes de software, en los nodos distribuidos. Los nodos son interconectados por una red, la cual proporciona un camino físico entre nodos (ver figura No 4.1). La conexión directa entre dos o más sistemas es algunas veces conocida como un enlace. Un sistema que ejecuta las funciones de la aplicación principal, y controla el sistema de comunicaciones es llamado algunas veces un host. En un sistema distribuido, el nombre de un objeto indica un sistema, proceso, o nodo; Una dirección indica dónde se encuentra el objeto nombrado, y una ruta nos indica cómo llegar allí.

Uno de los recursos compartidos más comunes en los sistemas distribuidos es la información. Muchas aplicaciones requieren distribuir cantidades de información entre diversos usuarios con diferentes facilidades de computación. Por medio de la distribución de la información, la confiabilidad puede mejorarse, haciendo replica de la información en copias múltiples. Los tiempos de acceso pueden mejorarse manteniendo copias locales de la información por medio de la replicación y división. Los sistemas de comunicación pueden ser usados para transmitir información y requerimientos de información entre distintos lugares, sistemas y programas. Mientras que los sistemas interconectados pueden o no formar un sistema de procesamiento distribuido por sí solos, los sistemas de comunicación de datos usados para intercambiar mensajes y datos, pueden considerarse por sí mismos como un sistema distribuido. De hecho, está físicamente distribuido, y sus componentes cooperan para proporcionar servicios comunes, y éstos son controlados por el sistema de administración de la red.

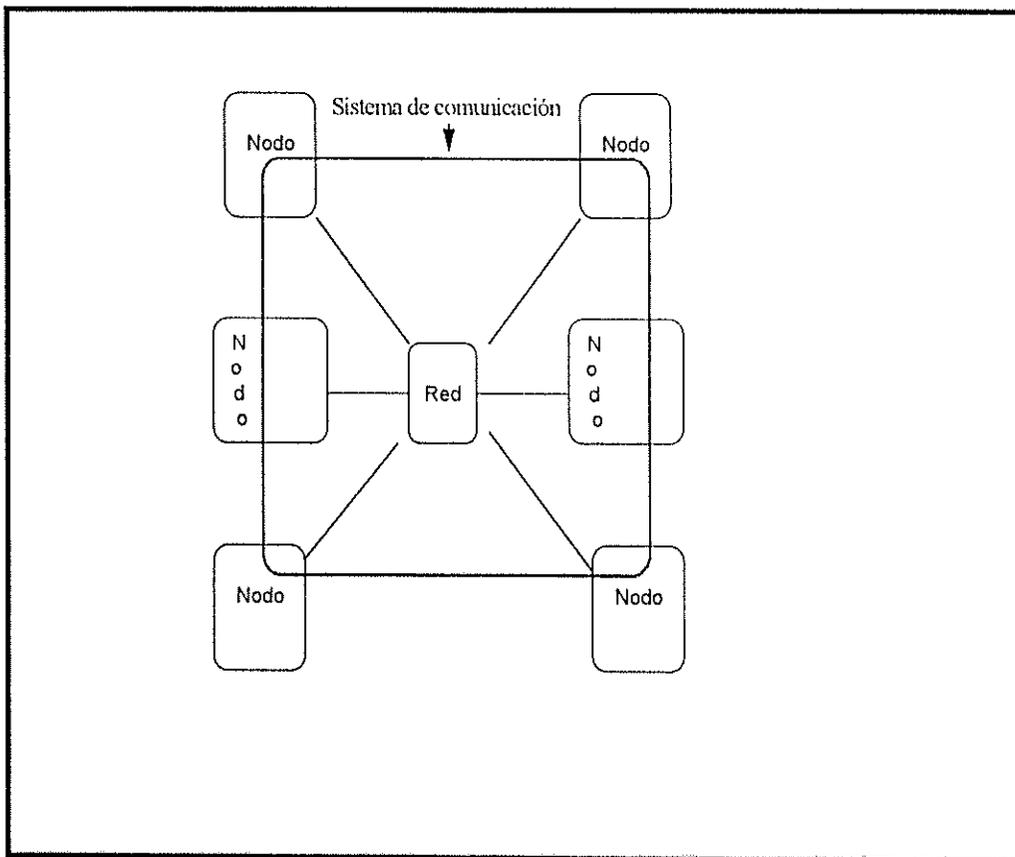


Fig. No. 4.1 Sistema de comunicación y red en un ambiente distribuido.

4.1.2 FUNCIONES DEL SISTEMA DE COMUNICACIÓN

Las siguientes son algunas de las funciones más importantes de los sistemas de comunicación.

Nombramiento y dirección. Un sistema de comunicación debe manipular nombres para objetos tales como: procesos, puertos, buzones, sistemas, y sesiones entre usuarios. Típicamente, los usuarios proporcionan estos nombres en una forma simbólica y el sistema de comunicación debe trasladar (mapear) el nombre simbólico a una dirección de la red de trabajo. Por lo tanto, un sistema de comunicación debe mantener tablas apropiadas de traslado (directorios). Los ruteadores están relacionados en trasladar las direcciones a su destino, mapeándolas en un camino o un objeto intermedio, tales como un gateway.

Segmentación. Si los mensajes o los archivos que van a ser transmitidos son muy grandes, el sistema de comunicación podría fragmentar un simple mensaje en múltiples segmentos (ver figura No.4.2). Una vez el mensaje es segmentado para la transmisión, la red de comunicación debe poder reensamblar lo fragmentado antes de ser liberado al usuario final.

Hay muchas razones para la segmentación :

- Los mensajes muy largos, incrementan el retraso del acceso para otros usuarios, ya que ellos tienen el control exclusivo de los recursos de la red de trabajo compartida por periodos más largo de tiempo.
- Mensajes pequeños mejoran la eficiencia y reducen el porcentaje de error de las transmisiones.
- Buffers internos, usados por los sistemas de comunicación, pueden limitar el tamaño del mensaje transmitido.
- Cuando varias redes de trabajo incluyen una ruta de transmisión particular, cada uno de los componentes de las redes pueden tener diferentes tamaños de mensajes.
- Cortando los mensajes grandes en pequeños segmentos, puede permitir el uso de enlaces de información en paralelo (en algunas redes), por lo tanto soportando transmisión en paralelo y reduciendo todos los retardos.

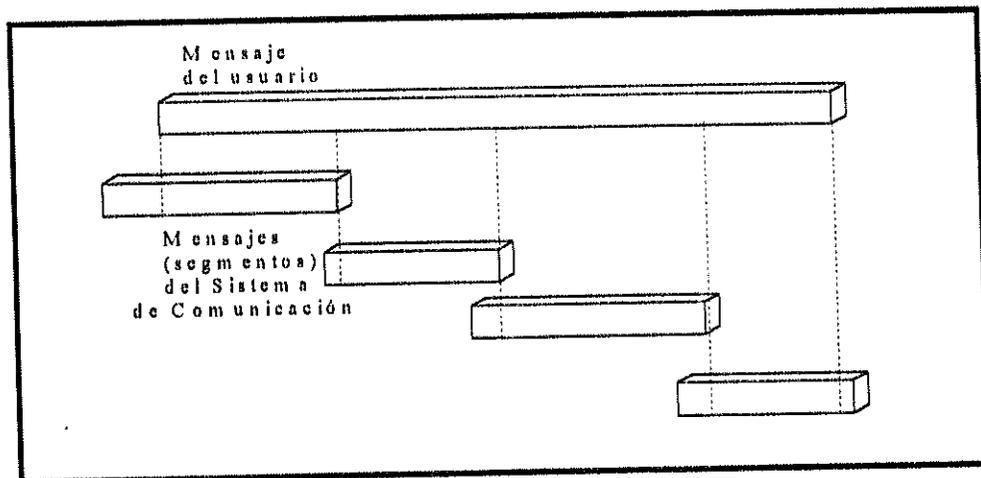


Fig. No. 4.2 Segmentación.

Bloques. Cuando un mensaje fragmentado es transmitido sobre la red, se incurre en una significativa sobrecarga en el protocolo, especialmente cuando el mensaje es corto. Por lo tanto, algunos sistemas de comunicación usan un protocolo de bloques, para combinar mensajes de diferentes usuarios dentro de un bloque para su transmisión.

Control de flujo. Muchas redes son diseñadas para compartir recursos limitados entre varios usuarios, asumiendo que no todos están activos al mismo tiempo. En realidad, la red de comunicaciones puede experimentar ciertos escenarios de tráfico pico, donde el tráfico resultante excede la capacidad de la red (la hora de congestión de tráfico es un buen ejemplo). La función del flujo de control de la red, es diseñada para prevenir el congestionamiento de la transmisión, los embotellamientos y optimizar el rendimiento de la red por la regulación del flujo de la información entre un par de entidades de comunicación. Por ejemplo, los protocolos de paso son utilizados para prevenir que el nodo remitente envíe más información de la que el receptor pueda manejar.

Sincronización. A fin de que un par de entidades puedan comunicarse entre sí, sus interacciones deben de estar sincronizadas. El receptor debe ser capaz de determinar el inicio y duración de cada señal, byte, bloque. De hecho, si un receptor es más rápido que el transmisor, éste puede mal interpretar información extra. Recíprocamente, el receptor puede perder alguna información si es más lento que el transmisor. Durante la comunicación, las entidades mantienen el estado de su información vía un protocolo de comunicación. Los estados de comunicación deben estar sincronizados en el tiempo de inicialización y después de una falla mayor. Los recursos compartidos tales como la información compartida, requieren sincronización entre procesos para soportar la integridad del recurso. Un ejemplo del protocolo de sincronización de multinivel utilizado en los sistemas de comunicación es el protocolo de comunicación avanzado de programa a programa de IBM (APPC).

Prioridad. Un sistema de comunicación puede asignar prioridad a los mensajes, para permitir el manejo preferencial cuando se compite por recursos. Los mensajes de alta prioridad (alarmas, alertas, interrupciones) deberfan tener retrasos más cortos. Un sistema de comunicación deberfa de ser capaz de asignar prioridad estáticamente o dinámicamente (de acuerdo con el contenido del mensaje o basados en la fuente o destino del mensaje).

Control de errores. Uno de los principales objetivos de la funcionalidad del sistema de comunicación, es proporcionar comunicación confiable y libre de errores. Las funciones de control de errores incluyen, detección, corrección y recuperación de errores. La detección de errores puede realizarse incluyendo información redundante al utilizar información de control, que permita la determinación de corrupción en la información, o asignando números secuenciales a los mensajes y detectando errores en secuencia. La información redundante puede compararse al determinar un posible error en caso de no coincidir. La información de control puede utilizar varios algoritmos, para calcular un dígito de chequeo, o una suma de chequeo de todos los bits de información. Al comparar los resultados comparados con aquellos recibidos, los errores pueden ser detectados. La secuencia es utilizada para determinar la pérdida, duplicación, o mensajes fuera de secuencia. La corrección de errores y recuperación usualmente se implementa a través de una retransmisión automática, posiblemente a través de una ruta diferente, aunque algunas veces los códigos de corrección de error pueden ser utilizados.

4.1.3 CAPAS, PROTOCOLOS, E INTERFACES.

Un sistema de comunicación es responsable de soportar comunicación entre nodos en un sistema distribuido. Permite a cualquier nodo de la red a comunicarse a cualquier otro nodo conectado en la red de comunicación. Las arquitecturas de las redes de trabajo computacional deberfan de facilitar la interconectividad entre sistemas homogéneos y heterogéneos. Para entender mejor cómo un sistema de comunicación es construido para soportar las comunicaciones, se considerara un modelo de comunicación típico de persona a persona. Tres niveles separados o capas, pueden ser definidos :

- La Capa cognoscitiva incluyen conceptos tales como, entendimiento, conocimiento, y existencia de símbolos compartidos mutuamente acordados. Por ejemplo, si una persona comunica a otra la descripción de un libro, entonces, por lo menos ambas personas deben de comprender qué es un libro.
- La Capa de lenguaje es utilizada para poner conceptos e ideas en palabras. Obviamente si las personas no tienen un lenguaje en común mutuamente acordado, cualquier comunicación será imposible.
- La Capa de Transmisión física proporciona los medios para la comunicación real, por ejemplo: ondas sonoras, papel para comunicación escrita, o cualquier otro medio.

Las tres capas son independientes cada una en su función, y las capas superiores requieren el soporte de las inferiores. Para comunicar ideas, el lenguaje y la transmisión son necesarios, pero lo contrario puede no ser cierto; los mismos principios deberfan de ser usados en la construcción de la arquitectura de sistemas de comunicación y sistemas distribuidos. Al menos, todas las arquitectura de redes de trabajo deberfan de compartir los mismos objetivos de alto nivel:

- Conectividad para permitir que varias plataformas de hardware y software estén interconectados en un sistema de red, con una imagen de sistema único, uniforme.
- Modularidad para permitir la construcción de diversos sistemas de trabajo, a partir de un número relativamente pequeño de componentes de objetivos generales.
- Confiabilidad para soportar comunicaciones libre de errores, con detección de errores y disponibilidad de corrección.
- Facilidad de implementación, uso y modificación; que proporcionen soluciones generales ampliamente aceptables para su instalación, modificación y administración de la red, para que brinden a los usuarios finales facilidades de comunicación de red transparente.

Para alcanzar estos objetivos de alto nivel, las arquitecturas de redes, deben de soportar un diseño altamente modular, donde cada función del modulo está organizada en capas funcionales, jerárquicas y arquitectónicas.

Las capas están compuestas de entidades, las cuales pueden ser componentes de hardware y/o procesos de software. Las entidades de la misma capa, pero en diferentes nodos de la red, reciben el nombre de entidades par; las capas del mismo nivel en diferentes nodos son capas par. La comunicación entre capas es gobernada por un conjunto de reglas o protocolos; los protocolos incluyen aunque no únicamente: formatos, un orden en el intercambio de la información, cualquier acción que se va a tomar en

la transmisión y recepción de la información. Las reglas y formatos para el intercambio de la información, a través de los límites entre dos capas adyacentes, comprenden un interface entre capas.

Por ejemplo, una arquitectura típica de sistema operativo puede contener las siguientes capas funcionales :

- La capa de aplicación es la de más alto nivel de la arquitectura. Típicamente, esta ejecuta la administración de los procesos de aplicación, distribución de información, comunicación interprocesos y descomposición de las funciones de aplicación en procesos distribuidos. La funcionalidad de la capa de aplicación se basa en capas de nivel inferior.
- La capa de sistema operativo distribuido proporciona servicios distribuidos en un sistema amplio, requeridos por la capa de aplicación. Esta soporta las funciones de: nombrar, direccionar, compartir recursos locales, protección, sincronización, intercomunicación y recuperación. El sistema operativo distribuido unifica las funciones distribuidas en una entidad lógica única y es responsable de la creación del sistema de imagen único (SSI).
- La capa de administración local y kernel soporta el sistema operativo distribuido en los nodos individuales. Asimismo, soporta comunicaciones interprocesos locales, memoria y accesos de entrada/salida, protección y multitarea. Esta capa soporta capas de más alto nivel, y proporciona los servicios solicitados por la capa del sistema operativo distribuido y se comunica con estas capas par (equivalentes) en otros nodos.
- La capa del sistema de comunicación soporta la comunicación requerida por las capas de aplicación, sistema operativo distribuido y administración local.

La arquitectura de capas proporciona varios beneficios importantes :

- Independencia de las capas. Cada capa está enterada de los servicios proporcionados por la capa inmediata inferior a ella, pero no de la implementación de las subsecuentes capas inferiores.
- Flexibilidad, un cambio de implementación en una capa (por ejemplo, nueva tecnología, hardware distinto, etc.), no debería de afectar a las capas inferiores y superiores a ella.
- Mantenimiento e implementación simplificada. Esto se debe al soporte de un diseño en capas modular y descomposición arquitectónica de la funcionalidad del sistema en general, en secciones más pequeñas y simples.
- Estandarización. Al encapsular la funcionalidad de la capa, servicios e interfaces en una entidad arquitectónica precisamente, permite a los estándares desarrollarse fácilmente.

4.1.4 MODELO DE REFERENCIA ISO.

La arquitectura de red de trabajo de sistemas IBM (SNA) es una arquitectura en capas que fue diseñada por IBM para las redes de comunicación de datos, basadas primordialmente en plataformas software y hardware compatibles e IBM. Sin embargo, otros proveedores y organizaciones ya han desarrollado o están en proceso de desarrollo de otras arquitecturas que pueden ofrecer métodos alternativos de construcción de redes de comunicación de datos. Por lo tanto, existen dos importantes tendencias que podemos encontrar en el mercado :

- Casi cada proveedor mayorista ofrece su propia arquitectura de red de trabajo, y estas arquitecturas no son compatibles.
- Si el cliente en un negocio utiliza múltiples plataformas de hardware para sus operaciones de procesamiento distribuido, la tarea de conectar redes de trabajo distintas se convierte más y más difícil.

Por lo tanto, la necesidad de un conjunto de reglas estándares o protocolos, para el intercambio de información entre arquitecturas de redes heterogéneas diferentes es urgente. Varias organizaciones de estándares están trabajando en el desarrollo de arquitecturas, protocolos e interfaces estándares. El Comité Consultor en Telegrafía y Telefonía Internacional (Consultative Committee on International Telegraphy and Telephony o CCITT), es una organización internacional de estándares, que desarrolló estándares para varios aspectos de las transmisiones de teléfono y datos es decir X.3, X.25, X.28, X.29, etc. El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), se centra en los estándares de las redes de trabajo. En particular, IEEE ha desarrollado estándares de redes de trabajo de área local, conocidos como proyecto 802.IEEE.

La suposición de que los estándares eran necesarios, para permitir la comunicación entre distintas plataformas, en conjunción con los beneficios y ventajas potenciales de las arquitecturas en capas, impulsó a la organización de estándares internacionales (ISO) a desarrollar un modelo de referencia, con el propósito específico de proveer bases comunes para los estándares de sistemas de comunicación. Este conjunto de estándares que actualmente están siendo desarrollados, recibe el nombre de interconexión de sistemas abiertos (OSI o Open Systems Interconnection). El modelo OSI ha adoptado las mejores características de las arquitecturas existentes. También proporciona características adicionales que hacen la interconexión de las redes heterogéneas más fácil y más eficiente de desarrollar.

Aun cuando el modelo OSI está aun siendo desarrollado, la arquitectura OSI usualmente es utilizada como un modelo de referencia por los proveedores y organizaciones estándares similares. No es sorprendente que sea una estructura de capas jerárquicas, la cual consiste de 7 capas arquitectónicas, (ver figura No. 4.3). Cada capa desempeña una función específica en la red, sirve a la capa inmediata superior, solicita servicios de la capa inmediata inferior, y se comunica sobre bases punto a punto, con las correspondientes capas en otros nodos.

A fin de entender mejor la implementación particular de varios sistemas de comunicación, a continuación se describirán brevemente las funciones básicas de las siete capas OSI.

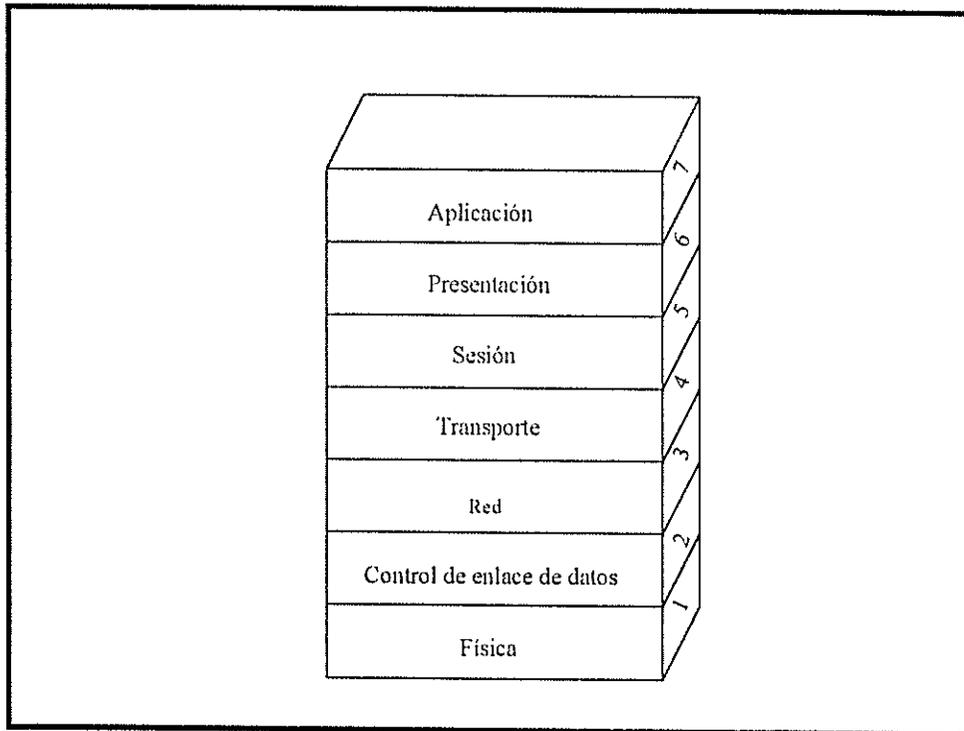


Fig. No. 4.3 Capas OSI.

CAPA FÍSICA OSI.

La capa física OSI, o capa 1, se encuentra en el extremo inferior del modelo OSI. Trata de la implementación física de una red de trabajo. La capa física OSI define las características eléctricas y de señales, que sean necesarias para establecer, mantener, y desconectar conexiones físicas en la red de trabajo. Actualmente la capa física del modelo OSI incluye los siguientes estándares desarrollados por el CCITT.

- V.35 y V.24 para transmisiones análogas.
- X.21 para transmisiones digitales.

La capa física OSI describe la manera en que los bits de información son enviados y recibidos, pero no proporciona reconocimiento de datos.

CAPA DE ENLACE DE DATOS OSI.

La capa de enlace de datos OSI, o capa 2, dentro de la estructura jerárquica, proporciona transmisiones sobre un enlace de datos individuales. La capa de enlace de datos soporta los siguientes servicios :

- Transporte de información desde un extremo de un enlace a otro.
- Activación y desactivación de un enlace de datos.
- Detección de errores de enlace de datos.
- Compartimiento de enlace de datos.
- Flujo de datos transparente.

- Recuperación y notificación de enlace de datos.

Entre los protocolos incluidos en la capa 2 del modelo OSI actualmente, se encuentran, the American National Standards Institute (ANSI), Advanced Data Communications Control Procedures (ADCCP), the International Standards Organization (ISO), High-Level Data Link Control (HDLC), Synchronous Data Link de SNA (SDLC), y los estándares ampliamente reconocidos llamados IEEE.802.2.

LA CAPA DE RED OSI.

La capa de red OSI, o capa 3, dentro de la estructura jerárquica, provee control entre dos nodos adyacentes. Las siguientes funciones han sido asignadas a la capa de red OSI.

- Direccionamiento de red (asignación, paridad y acoplamiento)
- Bloques y segmentación de unidades de mensajes
- Secuencia de datos
- Conmutación y ruteamiento
- Control de flujo local.
- Control de transferencia de datos expedito opcional.
- Control de detección de errores.
- Control de congestiónamiento.
- Notificación y recuperación de errores.

Todas estas funciones de la capa de red de OSI son incluidas en CCITT recomendando X.25 como un estándar internacional para intercambio de datos empaquetados en redes.

CAPA DE TRANSPORTE OSI

La capa de transporte OSI, o capa 4, dentro de la jerarquía, está definida para proporcionar el llamado control extremo a extremo, entre dos nodos usuarios. Esta capa toma paquetes de datos de la capa de red y los ensamble en mensajes. El modelo OSI define las operaciones de la capa de transporte como un procedimiento de múltiples fases.

- FASE 1 - Establecimiento. Esta fase establece la conexión de transporte a las capas inferiores.
- FASE 2 - Transferencia de datos. La fase de transferencia de datos transporta información entre las capas OSI.
- FASE 3 - Liberación. La fase de liberación desconecta la conexión de transporte.

CAPA DE SESIÓN OSI

La capa de sesión OSI, o capa 5, dentro de la jerarquía, proporciona funciones de redes relacionadas con las sesiones. Las siguientes funciones son definidas en el modelo OSI, para la capa de sesión.

- Inicialización y activación de sesiones.
- Finalización y liberación de sesiones.
- Sincronización y resincronización de conexión de sesiones.
- Control del diálogo.
- Transferencia de datos normal y expedita.

CAPA DE PRESENTACIÓN OSI

La capa de presentación OSI, o capa 6, dentro de la jerarquía, proporciona el protocolo para el flujo de la presentación de los datos. Estos protocolos deberfan de permitir a una red a determinar y mantener un registro (huella) de la información, como la sintaxis del mensaje, contenida en el flujo de datos del usuario. Las siguientes funciones han sido definidas para la capa de presentación OSI.

- Transformación de sintaxis de datos, es decir, encriptamiento de la información.
- Selección de la sintaxis de la presentación de imagen.

Los siguientes protocolos no están completamente definidos para esta capa :

- Terminal virtual
- Archivo virtual
- Transferencia y manipulación de jobs.

CAPA DE APLICACIÓN OSI

La capa de aplicación OSI es la capa de más alto nivel en la jerarquía del modelo OSI. Actualmente, la capa de aplicación es definida para contener todos los protocolos entre sistemas que no están incluidos en ninguna otra capa OSI. La capa de aplicación consiste en tres partes :

- Servicios de aplicación común. Son aquellos servicios que pueden ser usados por todas las partes de la comunicación. Los servicios de aplicación común proporcionan los protocolos que pueden seleccionar el tipo de conversación que mantienen dos usuarios, la estructura de está conversación, la estructura de transferencia de un archivo. Los servicios de aplicación común también proporcionan protocolos de control tales como: concurrencia, recuperación y commitment (comprometer).
- Servicios de aplicación específica. Son aquellos servicios que contienen los protocolos de todos los intercambios de información entre usuarios. Estos protocolos pueden ser privados dentro de una comunicación individual, o estándares de comunicación ampliamente aceptados, aun reconocidos internacionalmente. En estos momentos, los estándares de comunicación internacionales para la capa de aplicación OSI incluyen protocolos de mensaje X.400, protocolo de directorio X.500, y protocolo de administración, acceso y transferencia de archivos (FTMA).

- Elemento de usuario. En el modelo OSI, éste representa el punto final de la información del usuario (la organización y el punto de destino final). Si el elemento de usuario es definido, no habrá necesidad de una interface de usuario arriba de la capa de aplicación.

4.1.5 CONECTIVIDAD DE LOS COMPONENTES CLIENTE/SERVIDOR

En la arquitectura cliente/servidor, los sistemas cliente y servidor son construidos por un número de componentes interconectados. Cada componente provee una interface a través de la cual este se comunica con otro componente. Entonces, la comunicación entre los clientes y servidores puede ser vista como la comunicación entre elementos relevantes. Las dos clases distintas de componentes pueden ser definidas :

- Componentes de procesos. Típicamente, son los componentes de software, que desempeñan activamente algunas funciones de software.
- Componentes de recursos. Son aquellos que proporcionan servicios requeridos por los componentes de procesos.

Desde el punto de vista de las interacciones cliente/servidor, un componente de recursos activo actúa como un servidor, en cualquier lugar donde el usuario del componente de recurso y componente de procesos, actúen como clientes.

Los componentes de procesos y recursos, al igual que los clientes y servidores, entran en asociación con el propósito de comunicarse. Esta asociación es la conexión entre quien envía (requerimiento, mensaje, etc.) y el que recibe la información. Las conexiones cliente/servidor pueden ser estáticas o dinámicas. Una conexión estática es indicada en el momento de compilación, en el tiempo de carga, o en el tiempo de inicialización del sistema, y no pueden ser cambiadas. Las conexiones dinámicas pueden ser cambiadas en el tiempo de corrida.

4.1.6 COMUNICACIÓN Y SINCRONIZACIÓN

En el ambiente cooperativo distribuido cliente/servidor, la coordinación y la cooperación entre componentes (clientes y servidores) es proporcionada por el sistema de comunicación y las funciones de sincronización. Las funciones de comunicación involucran el intercambio de información, y son soportados por el flujo de control, control de errores, nombre, dirección, bloques y segmentación. Las funciones de comunicación involucran la coordinación de acciones entre dos o más componentes.

La comunicación y la sincronización están estrechamente relacionados. Cuando la comunicación es desempeñada en sistemas cercanamente acoplados de memoria compartida (tales como los sistemas multiprocesadores simétricos), los componentes de software tales como los semáforos o monitores son utilizados para la sincronización. En sistemas acoplados libremente más tradicionales que están interconectados por redes de trabajo de comunicación, mecanismos tales como pasar mensajes, deben ser utilizados para comunicación y sincronización.

En cualquier caso, los servicios de comunicación proporcionados por el sistema de comunicación pueden ser sin conexión u orientado a la comunicación. Los servicios sin conexión son aquellos donde cada transacción del mensaje es independiente de la previa o de las subsecuentes transacciones. Estos son servicios bajos-superiores relativamente simples de implementar. Un ejemplo es el servicio datagrama en donde no se le proporciona al usuario ninguna forma de respuesta a una transacción. Típicamente, los servicios datagrama son utilizados para emisión o transmisión de mensajes con múltiples destinos.

Los servicios orientados hacia la conexión proporcionan una relación entre la secuencia de unidades de información transmitidas por una capa de comunicación particular. Ese servicio es más complejo que el servicio de comunicación sin conexión. La mayoría de los servicios orientados a la conexión sufren tres fases de operación: establecimiento, información, y terminación. Una fase de establecimiento puede ser utilizada para negociar las opciones de conexión o sesión y, por ende, calidad del servicio. Una sesión terminal hacia un computador remoto o protocolos de red de conmutación de paquetes (X.25), que son aceptados por el CCITT, son ejemplos de servicios orientados a la conexión.

Respecto a las conexiones en general, el flujo de la información puede ser unidireccional (de un transmisor a un receptor) o el más complicado bidireccional. Este último comprende retorno del mensaje y sincronización en la respuesta al requerimiento inicial.

Las comunicaciones bidireccionales son una forma esencial de comunicación para la arquitectura cliente/servidor. Un componente del cliente que requiere de un servicio que va a ser realizado por su servidor (posiblemente remoto), espera hasta que el resultado de este requerimiento (si es que hay alguno) sea regresado. Las interacciones cliente/servidor bidireccionales pueden ser proporcionadas por la implementación de la comunicación orientada al mensaje, implementada en protocolos requerimiento respuesta, tales como la unidad lógica de IBM 6.2 (LU6.2), o por comunicaciones orientadas a procedimiento, tales como las llamadas de procedimiento remoto (RPC).

COMUNICACIÓN ORIENTADA A LOS PROCEDIMIENTOS - RPC

La comunicación orientada a los procedimientos permite a las aplicaciones en ambientes computacionales distribuidos, tal como, el Ambiente Computacional Distribuido de la Fundación del Software Abierto (DCE de la OSF), para correr sobre redes de trabajo heterogéneas; la tecnología básica que permite esta funcionalidad es la llamada de procedimiento remoto (RPC).

El modelo RPC se basa en la necesidad de correr componentes de procesos individual de una aplicación sobre cualquier sistema en la red. RPC utiliza una construcción de programación tradicional la llamada de procedimientos, cuyo uso se extiende desde un sistema individual hasta una red de sistemas. En el contexto del papel que desempeña un sistema de comunicación en un sistema cliente/servidor, un RPC requiere de un servicio en particular desde un componente de recurso (servidor), éste es emitido por un componente de proceso (cliente). La ubicación del componente de recurso se encuentra escondida del usuario (cliente). Los RPC son altamente adaptables a las aplicaciones cliente/servidor proporcionando generalmente a los desarrolladores un número de poderosas herramientas que son necesarias para construir tales aplicaciones. Esta herramientas incluyen dos grandes componentes :

- Un lenguaje y un compilador que simplifican el desarrollo de aplicaciones cliente/servidor distribuidas, producen códigos de fuentes portátiles.

- Facilidad de corrida de tiempo que permiten a las aplicaciones distribuidas ejecutar sobre nodos heterogéneos, múltiples, haciendo por lo tanto que las arquitecturas de sistemas, y los protocolos de red de trabajo subyacentes, transparentes a los procedimientos de aplicación.

Entre las varias implementaciones de RPC y las propuestas que compiten por el papel de RPC estándar; el RPC DCE parece ser uno de los candidatos más fuertes y merece ser examinado más de cerca.

Para desarrollar una aplicación cliente/servidor distribuida, DCE confiable; el desarrollador crea la definición de interface, utilizando el Lenguaje de Definición de Interface (Interface Definition Language o IDL). La sintaxis IDL es similar al lenguaje ANSI C, y además contiene varias construcciones de lenguaje apropiadas para la red. Una vez que las definiciones han sido creadas, el compilador IDL las transfiere hacia las colas que tienen el cliente y servidor (ver figura No. 4.4). La cola en el sistema del cliente actúa como un sustituto para el procedimiento requerido del servidor; de manera similar, la cola de un servidor sustituye a un cliente. Las colas son necesarias para automatizar operaciones que de otra manera serían manuales; como copiar argumentos en y desde encabezados RPC, convertir datos como sea necesario, y llamar a una corrida RPC.

La corrida RPC debería poseer las siguientes características:

- Transparencia e independencia de los protocolos de la red de trabajo subyacentes.
- Soporte para transmisión confiable, detección de errores y recuperación de las fallas de la red.
- Soporte Multinivel para procesamiento concurrente paralelo, capacidad para manejar múltiples requerimientos simultáneamente; y reducen de esta manera el tiempo requerido para completar una aplicación.
- Portabilidad e interoperabilidad con varios ambientes de sistemas.
- Soporte para la integridad de recursos y seguridad de aplicaciones.

La implementación DCE de llamadas de procedimiento remoto (RPC) incluye semánticas especiales, tanto para la independencia del transporte de la red, como para la transparencia de la misma. RPC DCE incluye facilidad de pipes para eliminar aquellas limitaciones de recursos como memoria principal inadecuada. El estándar X.500 de ISO es utilizada para proporcionar servicios de directorios globales. RPC DCE utiliza autenticación Kerberos, para soportar servicio de seguridad y niveles asncronos para soportar procesamiento paralelo y concurrente.

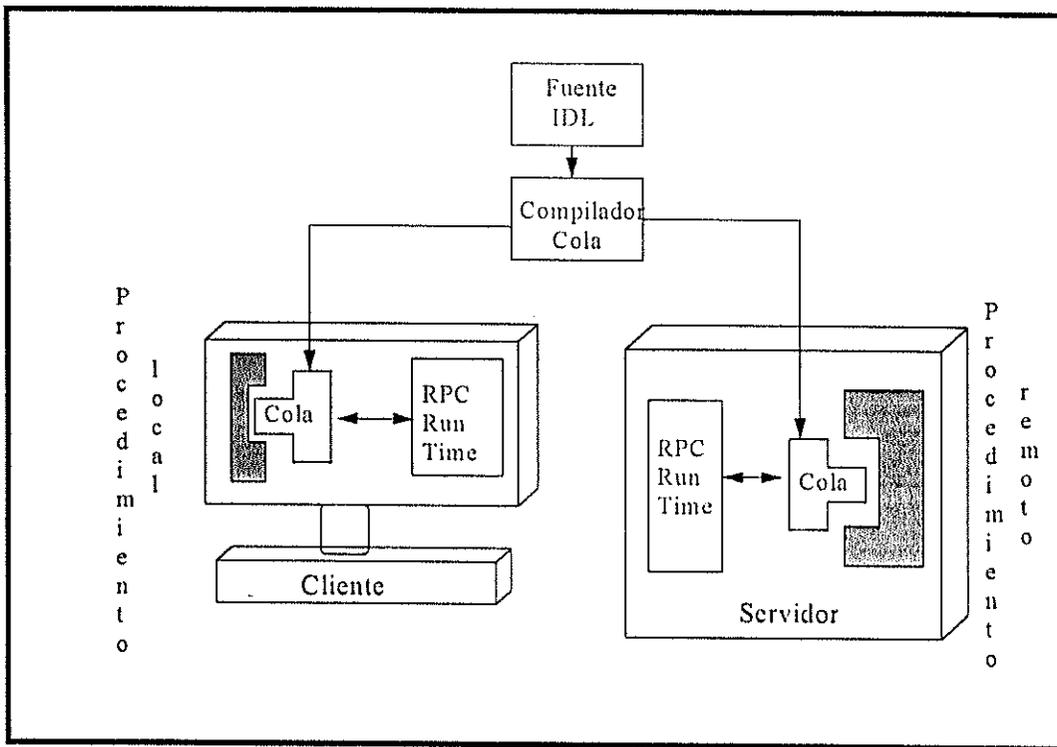


Fig. No. 4.4 Implementación RPC.

4.1.7 REDES DE TRABAJO

Los sistemas de comunicación son descritos como una colección de hardware y software, la cual soporta comunicación intersistemas e interprocesos entre los componentes de software en los nodos distribuidos.

Los actuales enlaces entre nodos comprendidos en una red representan una colección ordenada de caminos físicos entre los nodos interconectados.

A. CLASIFICACIÓN

Las redes de comunicación pueden clasificarse por la habilidad de un nodo en la red a comunicarse directamente con uno o más nodos.

- Comunicación punto a punto permite a un nodo comunicarse solamente con un nodo adyacente. En esta forma básica, una red punto a punto consiste en dos nodos conectados directamente (ver figura No. 4.5). De la misma manera, en una red de trabajo multinodo, en donde cada nodo está conectado a otro nodo adyacente, también forma parte de la red de trabajo punto a punto. En este caso, la adyacencia es medida por el número de pasos lógicos que toma llegar de un nodo a otro nodo adyacente. Por ejemplo, en una red Ethernet todos los nodos son un paso lógico para cualquier otro.

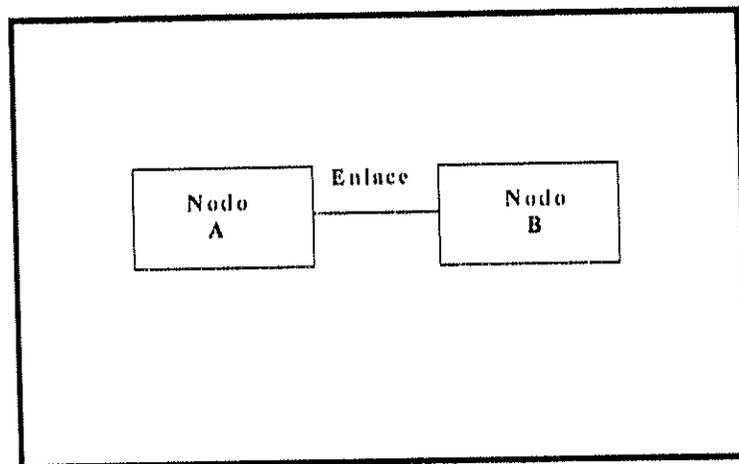


Fig. 4.5. Comunicación Punto a Punto.

- Las redes multipunto son redes de trabajo donde todos los nodos comparten una misma línea, y comparten tiempo en la línea. Las redes multipunto son similares a las líneas de teléfonos rurales, donde cada usuario levanta el teléfono y verifica si la línea está ocupada por cualquier otra persona; si es así, el usuario cuelga y prueba más tarde. Como el número de usuarios crece, es posible que el retraso o la demora llegue a ser grande. Por lo tanto, las redes multipunto son convenientes donde la transmisión de datos a alta velocidad no es necesaria, pero donde el costo de implementación es un factor que se debe considerar.

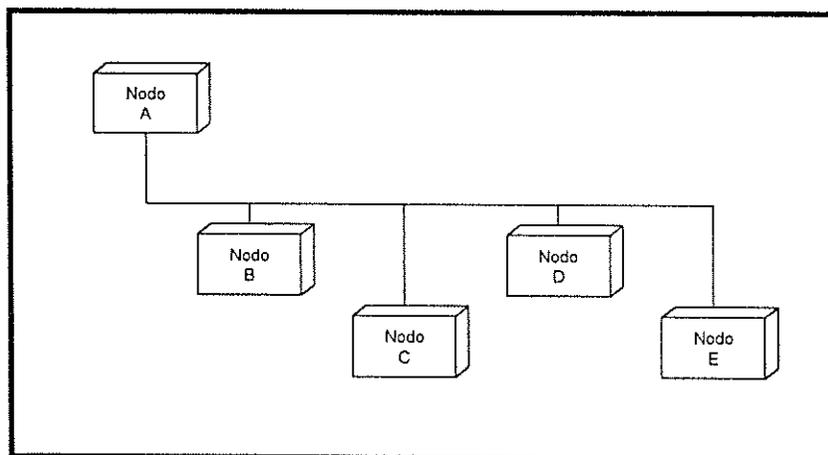


Fig. No. 4.6 Redes de comunicación Multipunto.

Las redes de comunicación pueden ser clasificadas por la manera en que los mensajes son transmitidos de un nodo a otro :

- Redes de difusión (Broadcast) (principalmente redes de área local) son aquellas donde todos los nodos están conectados a un medio común de transmisión; así, un simple mensaje puede alcanzar a todos los nodos.
- Redes de almacenamiento y reenvío (store and forward) (principalmente en redes de área extendida) son aquellas donde un mensaje completo es recibido en un buffer intermedio, antes de ser retransmitido hacia su destino. Los nodos en las redes de almacenamiento y envío están interconectados independientemente de las líneas de transmisión punto a punto.

B. TOPOLOGÍA

La topología de la red define la estructura de interconexión básica de nodos y enlaces. Las topologías pueden ser vistas como una gráfica arquitectónica de los nodos componentes de la red (cualquier sistema en una red) y los enlaces entre ellos. Los enlaces pueden ser líneas de teléfono, líneas privadas, canales de satélites, etc. Básicamente, los enlaces pueden ser divididos en enlaces físicos y enlaces virtuales.

Las redes de trabajo usan enlaces virtuales para permitir compartir la línea física a múltiples programas de la red y para la transferencia de datos. Las comunicaciones virtuales sobre líneas físicas son extremadamente valiosas ya que proporcionan capacidades de comunicación de costo-efectividad. Imagínese, por ejemplo, el costo de proveer una nueva línea física para cada nuevo programa de la red, si compartir los recursos vía enlaces virtuales no estuviera disponible.

- **REDES TOTALMENTE CONECTADAS:** cada nodo está conectado por un enlace dedicado punto a punto a cualquier otro nodo ver figura No. 4.7. Tal red es capaz de retardos pequeños y gran confiabilidad. La principal desventaja de tal topología es su alto costo. De hecho, n nodos requieren $n(n-1)/2$ enlaces, donde cada nodo debe tener $n-1$ conexiones. El agregar un nodo requiere que cada uno de los otros nodos debe ser modificado. Un caso especial de una red conectada completamente es una red malla. Esta tiene enlaces punto a punto entre algunos nodos. Una transmisión de almacenamiento y reenvío es requerida entre los nodos que no están directamente conectados.

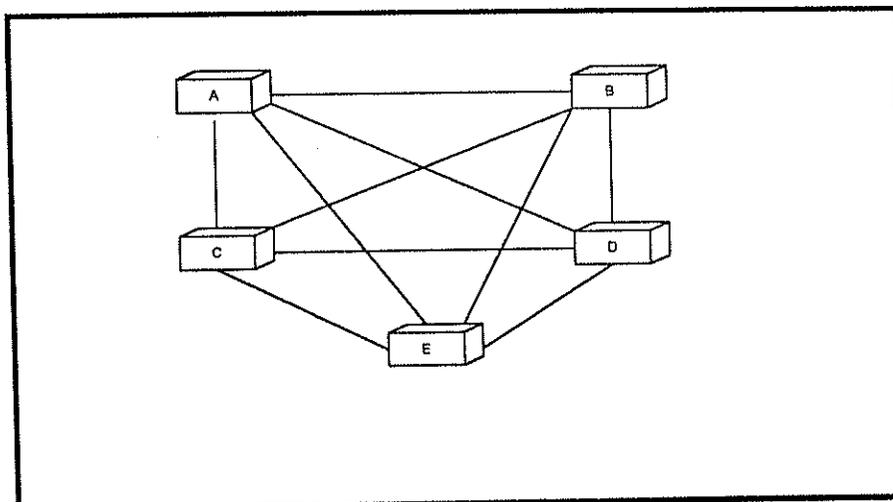


Fig. No. 4.7 Interconexión completa.

RED ESTRELLA: consiste de nodos que son conectados a una central de swicheo de nodos; esta topología es frecuentemente usada cuando múltiples terminales están conectadas a un computador central. Los costos de expansión de la red estrella son bajos, y el retardo no debería de exceder más de un nodo intermedio. La principal desventaja es la poca confiabilidad; si el conmutador central falla, detiene todas las comunicaciones.

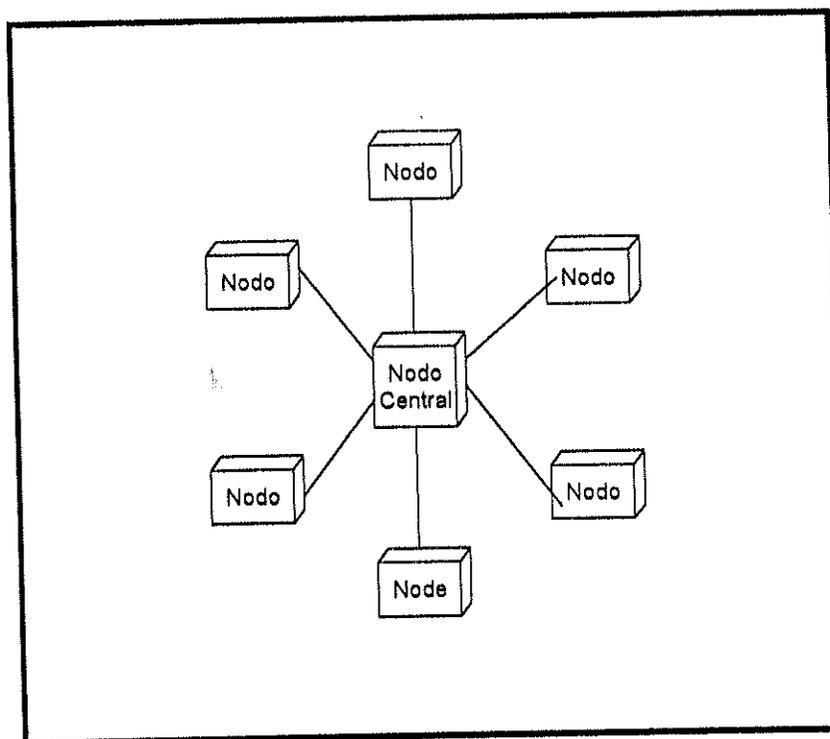


Fig. No. 4.8 Topología Estrella.

- **RED JERÁRQUICA O DE ÁRBOL:** es la topología donde un punto focal es ocupado por el host (computadora central), el cual actúa como maestro en una relación maestro-esclavo con los nodos de la

red de trabajo. Un ejemplo típico son las terminales de redes, en la arquitectura de redes de sistemas de IBM (SNA). Las redes de árboles son útiles para aplicaciones de control de procesos, ya que ellos reflejan la naturaleza jerárquica maestro-esclavo de un sistema de control.

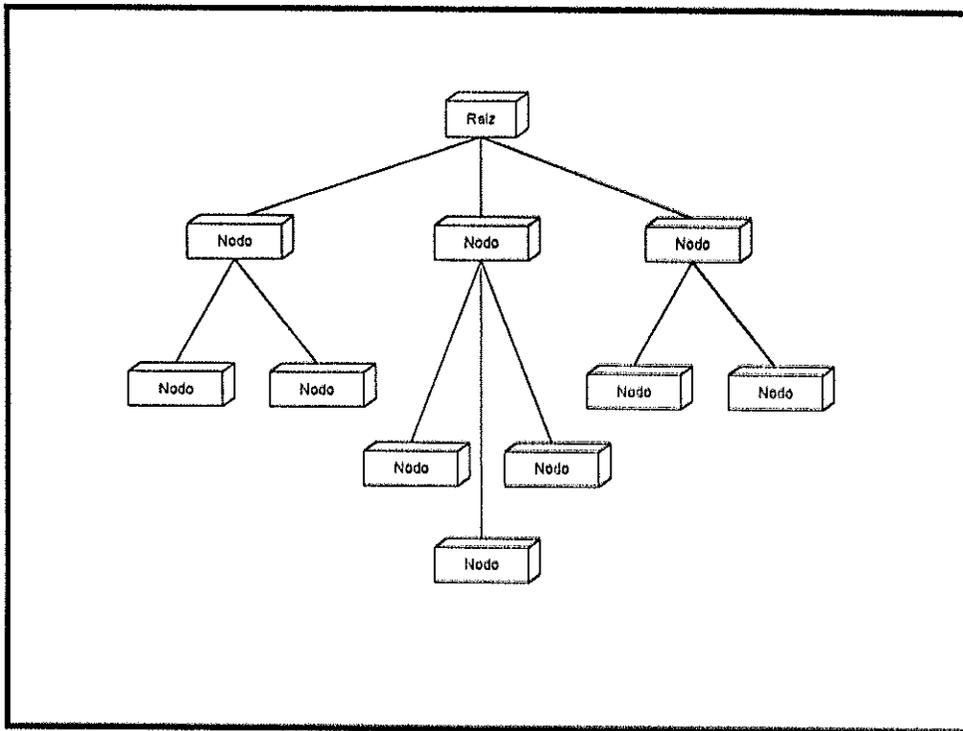


Fig. No. 4.9 Topología Jerárquica o de Árbol.

- **RED BUS:** es un ejemplo de la implementación de las redes de transmisión, donde el medio de comunicación compartido interconecta a todos los nodos. De aquí en adelante el tamaño de los enlaces y costos asociados son minimizados. Una simple interfaz es requerida para conectar un nuevo nodo en la red de trabajo, la cual resulta en costo de expansión bajo. La red Ethernet es un ejemplo de una red bus.

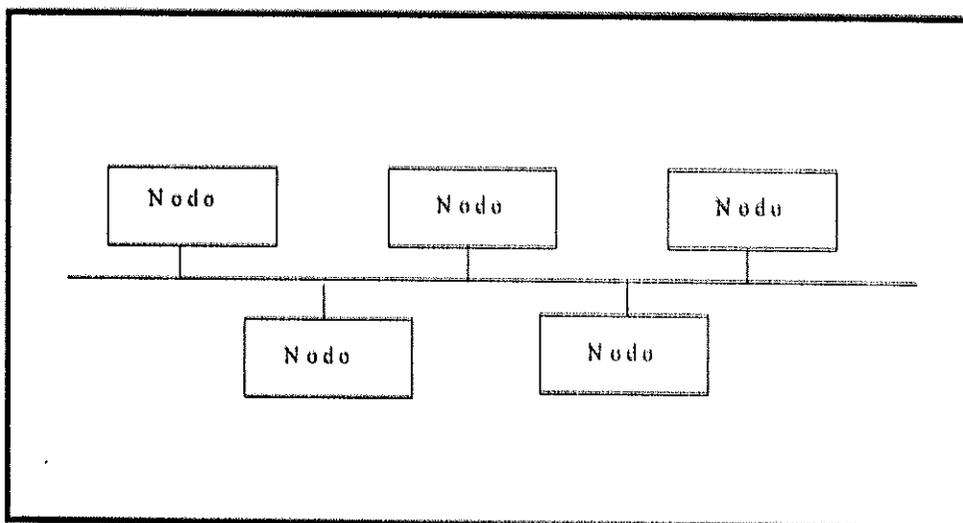


Fig. No. 4.10 Topología Bus.

Los estándares IEEE.802 describen protocolos que controlan los enlaces lógicos (IEEE.802) y los protocolos de acceso a los medios. La red bus como Ethernet son descritos por los estándares IEEE.802.3

- **LA RED ANILLO:** consiste en nodos unidos a sus vecinos por un lazo unidireccional. La transmisión puede ser por difusión (broadcast) o punto a punto. La señal es regenerada en cada interface, lo cual significa que la longitud del anillo no está limitada por la capacidad (como en la red bus). Solamente se requiere un enlace por cada nodo adicional, y el software de comunicación no requiere ruteadores. La red token-ring de IBM (IEEE.802.5) es un ejemplo de la topología anillo.

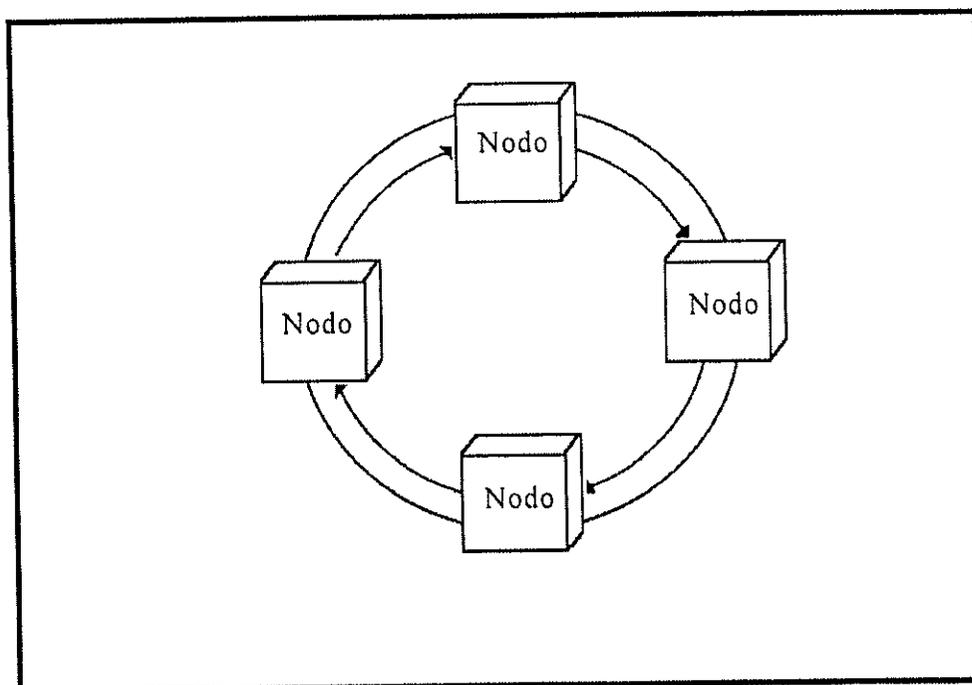


Fig. No. 4.11 Topología Anillo.

C. TÉCNICAS DE CONMUTACIÓN EN REDES

Las topologías de redes discutidas anteriormente ilustran que no todos los nodos tienen enlaces físicos directos entre ellos. Por lo tanto, una red debe proporcionar la función (realmente la función es en terminología OSI) de conmutar información entre los enlaces que proporciona el camino punto a punto entre algunos nodos.

Hay dos técnicas de conmutación usadas en las redes modernas : circuito conmutado y conmutación de paquetes.

Una red de circuito conmutado operan con conexiones (circuitos) dedicados entre dos nodos. Mientras el circuito está en su lugar, el emisor está garantizando que el mensaje será entregado a su destino. La capacidad de estos circuitos está garantizada, es decir, que ni otras redes activas podrían reducir la capacidad del circuito.

Las redes de conmutación de paquetes (usualmente, son usadas para conectar sistemas de computadoras) toman diferentes enfoques. Todo el tráfico es dividido en pequeños segmentos (paquetes) que

son mapeados (multiplexados) en intersistemas de alta capacidad, conectados para transmitir de un nodo a otro. Para implementar la conmutación de paquetes, los paquetes deben de llevar una identificación que permita al software del sistema operativo de la red (network operating system o NOS), enviar los paquetes a su destino. El hardware de la red entrega los paquetes a su destino; los paquetes son reensamblados por el software de la red.

La desventaja de la técnica de conmutación de paquetes es que al incrementarse la actividad, en un determinado par de compañeros de comunicación, reciben menor capacidad de recursos de la red. Esto es contrario al circuito conmutado; la capacidad disponible no está garantizada. Sin embargo un bajo costo y un hardware de la red de gran velocidad, proporciona un alto desempeño y una gran aceptación de las redes de conmutación de paquetes.

Algunas veces, la técnica de conmutación de mensajes está descrita como una opción tanto para la conmutación de circuito, como para la conmutación de paquetes. Estas involucran el almacenamiento de mensajes de cualquier longitud, incluyendo archivos, en el nodo de almacenamiento de conmutación. Los mensajes pueden ser almacenados por horas y días, hasta que el nodo destino desee recibir el mensaje. Este tipo de técnica de conmutación de paquetes es generalmente implementado en las aplicaciones de correo electrónico.

4.2 REDES DE ÁREA LOCAL

La proliferación de las PCs en las organizaciones, ha tenido como resultado a incrementar en el uso de las PCs, para una amplia variedad de funciones de negocios. Esto ha dado como resultado que se incremente la necesidad de comunicar las computadoras personales. La necesidad de comunicación incluye intercomunicación entre las computadoras personales, así como la comunicación con las facilidades de procesamiento de información centralizada y también con las fuentes de información de la corporación. Varias tecnologías de redes han estado desarrollando soporte para esta intercomunicación. Entre ellas tenemos: redes de área amplia, redes de área metropolitana y redes de área local.

4.2.1 CONCEPTOS DE REDES DE ÁREA LOCAL

Estaciones inteligentes, como computadoras personales o estaciones de trabajo basadas en sistemas UNIX, pueden ser usadas como sistemas solos, que soportan aplicaciones locales. Sin embargo, los ambientes de trabajo en grupo llegan a ser más y más populares; la razón para interconectar esas estaciones de trabajo inteligentes en una red llega a ser más evidente, entre ellos tenemos :

- La necesidad de acceder información almacenada en otro sistema no local.
- La necesidad de los miembros de trabajo en grupo a compartir dispositivos, que son muy caros para ser usados por un simple usuario (impresora láser).
- La necesidad de los miembros de trabajo en grupo a intercambiar información electrónica.

LAN, WAN Y MAN

Las distancias entre los usuarios de las redes es un factor que determina el tipo de requerimiento de la red, por lo tanto, la tecnología es fundamental. Una situación incluye los requerimientos de los usuarios, a acceder capacidades de procesamiento y almacenamiento de información, típicamente disponibles en los mainframes. Similarmente, la interconectividad puede ser requerida por los usuarios que geográficamente están ampliamente dispersos. En este caso las soluciones de redes, pueden incluir e involucrar, telecomunicación pública para facilitar el intercambio de la información. Las redes que unen a todos esos usuarios son llamadas redes de área extendida (Wide Area Networks WAN).

Algunas veces, esto es usado para diferenciar las redes de área amplia que pueden tener localidades remotas que miden miles de kilómetros (usuarios en New York, Los Ángeles y Tokio) y las redes que unen usuarios dentro de una área metropolitana en particular. Las redes que operan dentro de una ciudad o aquellas que usan facilidades de telecomunicaciones físicas, típicamente asociadas a la infraestructura de la ciudad (sistemas de cableado bajo tierra), son llamadas redes de área metropolitana (Metropolitan Area Network MAN), una típica MAN provee comunicaciones de voz, datos y video a una velocidad 45-600 Mbits /s (un millón de bits por segundo) a distancias alrededor de 1 a 75 Kilómetros.

Las comunicaciones en distancias relativamente pequeñas entre estaciones de trabajo inteligentes (como computadoras personales) son soportados por la tecnología de redes, lo que se conoce como las redes de área local (Local Area Network LAN). El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) define a las LAN como sistemas de comunicación de datos que permiten a un número de dispositivos independientes a comunicarse directamente con cualquier otro, dentro de una área geográficamente moderada sobre canales de comunicación físicos de una velocidad de transmisión de datos moderada.

Las redes de área local pueden ser usadas para compartir datos, accesos a dispositivos, correo electrónico, monitoreo de procesos en ambientes de fábricas, y aun para alarmas y sistemas de seguridad. Sin embargo, la característica más interesante de las redes de área local es la habilidad para soportar aplicaciones cooperativas cliente/servidor en las cuales las aplicaciones se ejecutan parcialmente en la estación de trabajo de la red LAN, y parcialmente en el servidor de la red LAN y aun en host mainframes. También el rango de aplicación de las LAN puede ser ampliamente significativo por la interconexión de varias redes de trabajo. Las interconexiones LAN pueden implementarse sobre las redes de área amplia, así extendiendo las capacidades de expansión de las redes de área local más allá de las limitadas distancias tradicionales de una típica red LAN. Al mismo tiempo, la facilidad de uso y de interconexión directa típica de las LANs todavía se mantienen.

CARACTERÍSTICAS Y COMPONENTES DE LAS LANs

La definición de IEEE de LAN proporcionan características que distinguen las redes de área local de otras tecnologías de redes que incluyen :

- Por la independencia permitida de los dispositivos a comunicarse directamente uno con otro. LAN soporta comunicación punto a punto entre nodos. Esto está en contraste con las centrales que controlan sistemas jerárquicos, por ejemplo la Arquitectura de Sistema de Red (SNA) de IBM.
- Por el énfasis de un moderado tamaño del área geográfica. El IEEE separa a las LAN de las redes de área extendida. Típicamente, una LAN no excede una distancia de alrededor de 10 kilómetros, y

frecuente está limitada por una simple construcción o simples construcciones en lugares relativamente cercanos entre sí, por ejemplo, un campus universitario.

- Por la definición del canal de comunicación físico que transmite datos a una velocidad moderada. IEEE contrasta las LAN con las WAN, las cuales usan frecuentemente las facilidades de comunicación pública.

Una moderada velocidad de transmisión de datos, es usada para implicar la velocidad de transmisión de datos en una LAN, que llega a ser más lenta que todos los enlaces directos en los mainframes y comunicación de canal a canal, medidas en varios millones de bits por segundo. Sin embargo, con los avances en tecnología de transmisión física y especialmente con la llegada de la comunicación con fibra óptica, las redes de área local pueden soportar velocidades de transmisión de 1 a 100 Mbits/seg.

Una red de área local típica, que corresponde a la definición de LAN de IEEE, contiene dos tipos generales de componentes: nodos y enlaces entre nodos. En la terminología LAN, los nodos pueden ser cualquier dispositivo conectado a la red, son conocidos como una estación. Todas las estaciones en las LANs están enlazadas o interconectadas vía sistemas de cable, los cuales incluyen canales de comunicación físico (alambre, cable) y cualquier otro dispositivo necesario para conectar las estaciones a la red.

Por ejemplo, para evitar la pérdida de la señal sobre el cable, se instalan regeneradores de señal o repetidoras en las LANs. Cada estación debe de poseer suficiente inteligencia para manejar las funciones de control de comunicaciones. Así, los dispositivos periféricos (como impresoras y discos duros) no son estaciones LANs, pero son conectados sobre las estaciones inteligentes.

Las redes, incluyendo las redes de área local, se caracterizan por la forma del sistema de cableado topología de la red. Adicionalmente, diferentes redes de área local son caracterizadas por :

- El medio de transmisión, el tipo de cable que es usado en la red LAN.
- La técnica de transmisión, que determina cómo el medio de transmisión es usado en las comunicaciones.
- El método de control de acceso, es el método por el cual las estaciones LAN controlan su acceso al medio de transmisión.

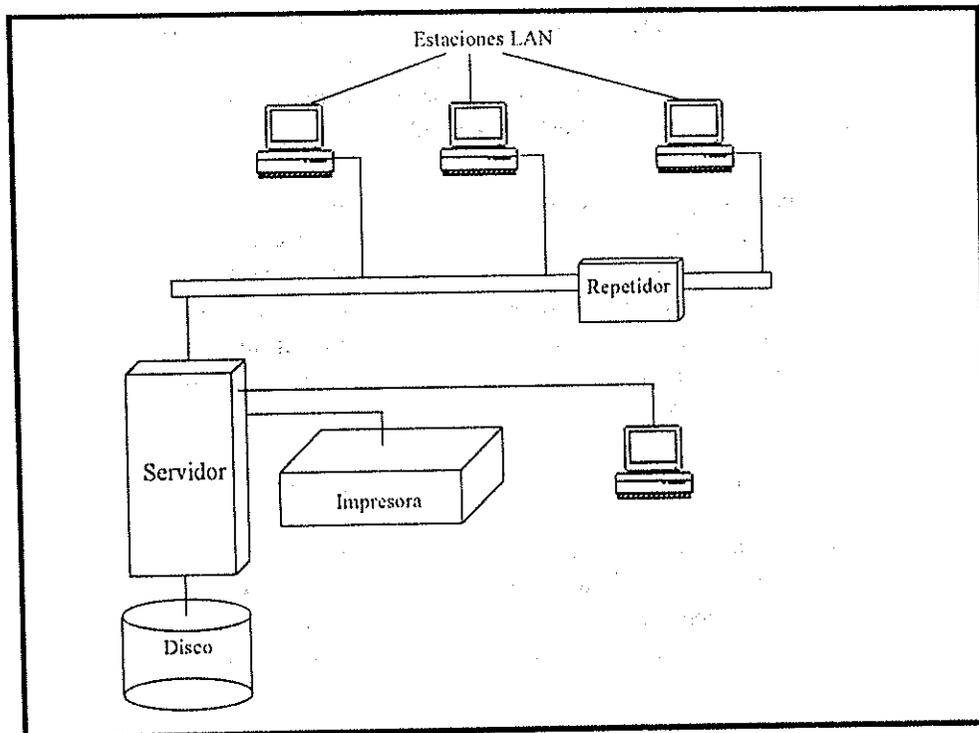


Fig. No. 4.12 Componentes LAN

4.2.2 TRANSMISIÓN Y MÉTODOS DE CONTROL DE ACCESO

La transmisión física de información sobre las redes de área local puede ser descrita en dos categorías principales: el medio actual usado para la transmisión y el camino que este medio usa.

MEDIO DE TRANSMISIÓN

Una amplia variedad de enlaces de comunicación física son usadas para la transmisión de la información. La industria de la comunicación continúa expandiendo los medios de transmisión disponibles para llegar a una velocidad extremadamente alta mientras se mejoran los costos y se dispone de estas nuevas conexiones.

Aun los medios tradicionales empleados en aplicaciones de telecomunicación convencional, pueden ser usados en las redes de área local; actualmente las LANs son frecuentemente implementadas usando enlaces con parejas de alambres, cable coaxial y fibra óptica.

Un par de alambre trenzado consiste en dos cables de alambre de cobre entrelazados y aislados. Usualmente, tales parejas forman un cable agrupados en parejas y encerrados en una envoltura protectora común. Un cableado telefónico típico en un edificio es un ejemplo de este tipo de cable. El bajo costo relativamente y la alta disponibilidad de este cable, tiene como resultado un gran popularidad del alambre par trenzado en las implementaciones para las redes de área local.

Para eliminar las posibles interferencias eléctricas, el alambre par trenzado es encerrado en un protector especial de alta calidad. Este cable es llamado cable par de alambre trenzado protegido, y es usado con alta confiabilidad y alta velocidad de transmisión sobre largas distancias donde es requerido. El sistema de cableado IBM tipo 1 y 2 son ejemplos del cable de alambre par trenzado.

El cable coaxial es familiar a los usuarios de la televisión y especialmente aquellos con TV con cable. El cable coaxial contiene un conductor central (usualmente cobre) que está rodeado por material aislante, y otros conductores (malla de alambre entrelazado), y otra envoltura protectora aislante y flexible, aunque es más costoso; el cable coaxial está mejor aislado de interferencias eléctricas que el alambre par trenzado. El cable coaxial puede soportar velocidades de transmisión de 100 Mbits/Seg. Los sistemas de comunicación DEC conectados con cables delgados y estándares Ethernet son ejemplos de cables coaxial usados en las implementaciones LANs.

Los enlaces con fibra óptica son relativamente un nuevo medio de transmisión disponible para implementaciones LAN comerciales. La fibra óptica contiene en su parte central un cilindro de vidrio extremadamente delgado rodeado por una capa concéntrica de vidrio llamado revestimiento. Las señales ópticas toman forma de rayos de luz modulados que atraviesan longitudes de los enlaces de fibra óptica a la velocidad de la luz. El revestimiento es hecho de material cuyo índice refractor es más bajo que el índice de la parte central. Por lo tanto, las señales de luz que viajan a través de la parte central son reflejadas desde el reverso del revestimiento hacia la parte central. Un número de fibras ópticas están sujetas unas con otras dentro de un cable de fibra óptica, envueltos por una cubierta protectora. Los cables de fibra óptica están caracterizados por su peso ligero comparado con el cable coaxial, pero con un costo significativamente grande. Sin embargo, las señales de luz transmitidas sobre el cable de fibra óptica no están sujetos a interferencias eléctricas. El medio de transmisión óptica puede soportar velocidades de transmisión extremadamente altas, velocidades de transmisión alrededor de 565 Mbits/Seg. pueden estar disponibles en sistemas comerciales, experimentos han demostrado que velocidades de transmisión de información alrededor de 200,000 Mbits/Seg. El sistema de cableado tipo 5 es un ejemplo de cable de fibra óptica usado por las redes de computadoras.

TÉCNICAS DE TRANSMISIÓN

Cualquiera que sea el medio de transmisión empleado en un ambiente LAN, el diseñador de la red LAN debe seleccionar una técnica que utilizara para transmitir señales sobre los enlaces de comunicación física. En general, existen dos técnicas de transmisión disponible para transmisión sobre un canal de comunicación físico : la banda base y la banda ancha.

La transmisión de banda base utiliza señales discretas (pulsaciones de electricidad o luz que representan un 0 y 1 binarios) para transportar información sobre el medio de transmisión físico. Tal técnica de señalización recibe el nombre de señalización digital. La transmisión de banda base utiliza toda la capacidad del canal de comunicación para transmitir una señal de información individual, ya que múltiples estaciones conectadas a la red de trabajo deben de compartir un canal de comunicación común; una técnica conocida como multiplexación por división del tiempo (TDM) es usualmente empleada para permitir a las estaciones conectadas a transmitir una señal a la vez. Una de las preocupaciones acerca de la transmisión digital es que conforme una señal viaja a través de un canal la misma puede debilitarse; su forma se distorsiona y la estación receptora puede tener dificultad en reconocer e interpretar la señal. En otras palabras, la diferencia entre 0 y 1 puede ser insuficiente para distinguir entre ambos. Dispositivos de repetición especiales son colocados estratégicamente a lo largo de la ruta de una señal para solucionar el

problema de deterioro de la señal. Los repetidores regeneran totalmente la señal y la transmiten hacia su destino.

La transmisión de banda ancha generalmente emplea señales no discretas (análogas) que son continuamente transmitidas sobre un medio de transmisión en forma de ondas electromagnéticas. La información discreta es encriptada en ondas análogas utilizando modulación de amplitud, frecuencia o de fase de la señal base (transportador). En transmisión de radio, la modulación de amplitud es conocida como AM y la modulación de frecuencia es conocida como FM. En general, mientras más grande sea la frecuencia de la señal transportadora más alto será el volumen de la información que pueda ser transportado por esta señal. La diferencia entre las frecuencias más altas y bajas que son transportados sobre los reflejos de un canal, es la capacidad de transporte de información del canal y recibe el nombre de amplitud de banda del canal. La amplitud de banda está directamente relacionada con otra medida de capacidad de canal, el número de bits/seg. que pueden ser transportados por el canal conocido como tasa de información.

Similar a la multiplexación por división del tiempo de transmisión de banda base, la transmisión de banda ancha análoga emplea multiplexación por división de frecuencia (FDM), la cual divide la amplitud de banda disponible en múltiples canales de comunicación. Alguno de estos canales puede ser usado para transmisión de información, mientras que otros pueden ser empleados para transmisión de video, fax, teléfono, simultáneamente. La mayoría de información transmitida en ambientes LANs abarca señales discretas compuestas de información digital. Cuando una transmisión de banda ancha utiliza las señales digitales, éstas deben de ser convertidas a la forma análoga modulando la señal transportadora análoga. En la estación receptora, estas señales deben de convertirse de vuelta (desmoduladas) a la forma original digital.

El proceso de modulación-desmodulación es realizado por dispositivos conocidos como módems (modulador-desmodulador). Varios esquemas de encriptamiento son empleados por LAN para representar 0 y 1 discretos. Algunos de los esquemas de encriptamiento mejor conocidos son, el RS-232-C de Electronic Industry Association y el encriptamiento diferencial complementado a cero utilizado en el control de enlaces de datos sincronicos de alto desempeño de IBM (SDLC).

CONTROL DE TRANSMISIÓN

En general, se puede clasificar varios métodos de control de transmisión como sigue :

- El control centralizado es aquel en el cual una estación controla la red de trabajo completa y otorga a otras estaciones permisos para transmitir.
- El control Aleatorio permite a cualquier estación LAN transmitir sin contar con un permiso específico.
- El control distribuido otorga el derecho de transmitir solamente a una estación a la vez. El derecho de transmitir se pasa de una estación a la siguiente. Todas las estaciones cooperan para controlar el acceso a la red de trabajo.

Cada uno de estos métodos de control de transmisión ofrece sus propias ventajas y desventajas y posee métodos de control de acceso diseñados específicamente para trabajar mejor con un control de transmisión en particular. Por ejemplo, el control de transmisión centralizado proporciona mayor facilidad en la coordinación y administración de la red de trabajo, y requiere interfaces simples de estación a la red. Al mismo tiempo, el control de transmisión centralizado, por definición, proporciona un punto individual de

falla, y el punto de control central puede probar ser un cuello de botella. Los métodos de control de acceso están diseñados para facilitar y emplear varios controles de transmisión.

El control centralizado puede emplear los siguientes métodos de control de acceso :

- **Polling⁹.** Una estación (maestro) envía una notificación a todas las otras estaciones (secundarias) indicando que una estación determinada cuenta con permiso para transmitir. Si la estación que recibe la notificación (polled) tiene un mensaje que transmitir, ésta envía el mensaje a la estación maestra. Seguidamente, la estación maestra dirige el mensaje a su destino. Mientras la estación notificada (polled) está siendo escuchada, una estación secundaria es autorizada para transmitir más de un mensaje.
- **Conmutación de circuitos.** Puede ser utilizado con éxito en una LAN de control centralizado, implementada a través del uso de una topología estrella. Aquí, una estación central recibe requerimientos de transmisión de una estación secundaria y establece una conexión (circuito) entre el emisor y su receptor deseado. La conmutación de circuitos es usada ampliamente en telefonía, especialmente en PBX.
- **Acceso múltiple de división del tiempo (TDMA).** Proporciona un slot de tiempo específico para cada estación en la red de trabajo. La estación cuenta con permiso para transmitir durante una cantidad de tiempo. El ciclo de tiempo es inicializado y sincronizado por una estación maestra. TDMA puede ser utilizado con éxito con una topología de LAN de bus.

Una de las técnicas de control de acceso mejor conocidas para el control de transmisión aleatorio, es el acceso múltiple por detección de portador con detección de colisión (CSMA/CD). Este es uno de los métodos de acceso LAN más comúnmente usados; es empleado por Ethernet, y también se define como uno de los estándares IEEE LAN. En CSMA/CD antes de que una estación pueda transmitir, ésta debe escuchar que la red determine si otra estación está transmitiendo o no (percibir el transportador). Si la red de trabajo está silenciosa, la estación puede transmitir su mensaje, el cual llega a toda otra estación. Solamente aquellas estaciones que están indicadas en el mensaje recibirán en realidad el mensaje.

Es concebible, sin embargo, que dos o más estaciones pudieran transmitir sus mensajes simultáneamente, resultando una colisión de mensaje. Si esto sucede, las estaciones receptoras ignoran los mensajes mutilados, mientras que las estaciones transmisoras intentan retransmitir los mensajes. Para evitar repetidas colisiones, cada estación transmisora espera un período de tiempo (determinado por un número aleatorio generado por la estación) antes de transmitir nuevamente. En tráfico liviano de mensajes CSMA/CD es muy eficiente, y el acceso a los medios de transmisión es rápido. El tráfico pesado de mensajes, sin embargo, lleva a un número creciente de colisiones. Como resultado, la eficiencia de los métodos de acceso y el desempeño de la red de trabajo se deteriora.

El control de transmisión distribuido puede ser soportado por métodos de control de acceso como aquellos bien conocidos como pase de testigo (tokens) y acceso múltiple por detección de portación con evasión de colisión (CSMA/CA).

El traslado de token ring (pase de testigo en anillo) es ampliamente utilizado en redes de trabajo con topología de anillo (por ejemplo, token ring de IBM). Un token es un pequeño mensaje que está constantemente circulando alrededor del anillo. El pase de token puede ser utilizado en una topología de

⁹ Polling : Registro, Notificación.

árbol o de bus. Los métodos de token bus son similares al token ring, emulando el método de token ring en un nivel de topología lógica.

Cuando un token está señalado como "libre", es recibido por una estación y la estación puede transmitir un mensaje. Cuando una estación transmite un mensaje, éste agrega el mensaje con un token marcado como "ocupado". Cada estación que debería de recibir el mensaje lo copia y actualiza los bits del token para indicar si la estación recibió o no el mensaje. Finalmente, el mensaje regresa al emisor. La estación emisora reinicializa el token para liberar y remover el mensaje. El pase de token garantiza que cada estación tiene la oportunidad de transmitir un mensaje durante un periodo de tiempo predeterminado. La principal desventaja de la técnica de traslado de token, es su complejidad y la sobrecarga de procesamiento token.

CSMA/CA es muy similar a los métodos de acceso CSMA/CD. Cada estación escucha al transportador mientras que una transmisión desde una de las estaciones está en progreso (si hay alguna). Cuando la línea está libre las estaciones comienzan a transmitir sus mensajes en el orden de sus prioridades que les fueron asignadas. A fin de evitar potenciales colisiones de mensajes, cada estación espera un período de tiempo específico antes de comenzar la transmisión.

Al diseñar redes de trabajo de área local, muchos factores interdependientes deben de ser tomados en consideración: los medios de transmisión, métodos de control de transmisión y acceso, topología de red de trabajo, ancho de banda, y tasas de información. Todos estos factores afectan el desempeño y costo de la red de trabajo. Las decisiones concernientes con la topología de red de trabajo, control de transmisión y métodos de control de acceso deberían de tomarse basados en los requerimientos de procesamiento y costo de un LAN particular.

4.2.3 ARQUITECTURA IEEE 802

La IEEE juega un papel importante en la definición de los estándares de las LAN. Especialmente, IEEE se ha encargado de un proyecto en especial, el proyecto 802, que tiene como meta el desarrollo de una arquitectura de red flexible orientada especialmente a las redes de área local.

ESTANDARES LAN Y EL PROYECTO IEEE 802

IEEE ha intentado definir la arquitectura LAN en convenio con la arquitectura de interconexión de sistemas abiertos de la ISO, descrito anteriormente. Sin embargo, el alcance del proyecto IEEE 802 es limitado solamente por las capas OSI inferiores, la capa física y la capa de enlace. Las funciones desempeñadas por las capas OSI superiores son relegadas a los proveedores de LAN y aun a los usuarios. Esas funciones son proporcionadas por el sistema operativo de la red (NOS o network operating system), NOS está típicamente implementado como un producto de software que es capaz de desempeñarse como servidor de impresión, servidor de archivos y soporte de correo electrónico (funciones de las capas OSI superiores de aplicación), conectividad entre redes y soporte de administración de redes (funciones de las capas sesión, transporte y red).

Ya que un estándar individual capaz de llenar todos los requerimientos LAN era muy difícil de definir, la aproximación dada por el proyecto IEEE 802, fue hecho para desarrollar conjuntos, o familias de estándares. Esos conjuntos de estándares están organizados en una jerarquía, como tres conjuntos de capas

que corresponden a las capas de la arquitectura LAN. El proyecto IEEE 802 define un estándar individual para la capa de enlace lógico: IEEE 802.2. El estándar de la capa física incluye par de alambre trenzado, cable coaxial y cables de fibra óptica, conjuntamente con los tipos de transmisión, métodos de encriptamiento y tasa de información. El control de accesos al medio y la capa física son descritos por múltiples estándares IEEE 802. Algunos de ellos son listados a continuación.

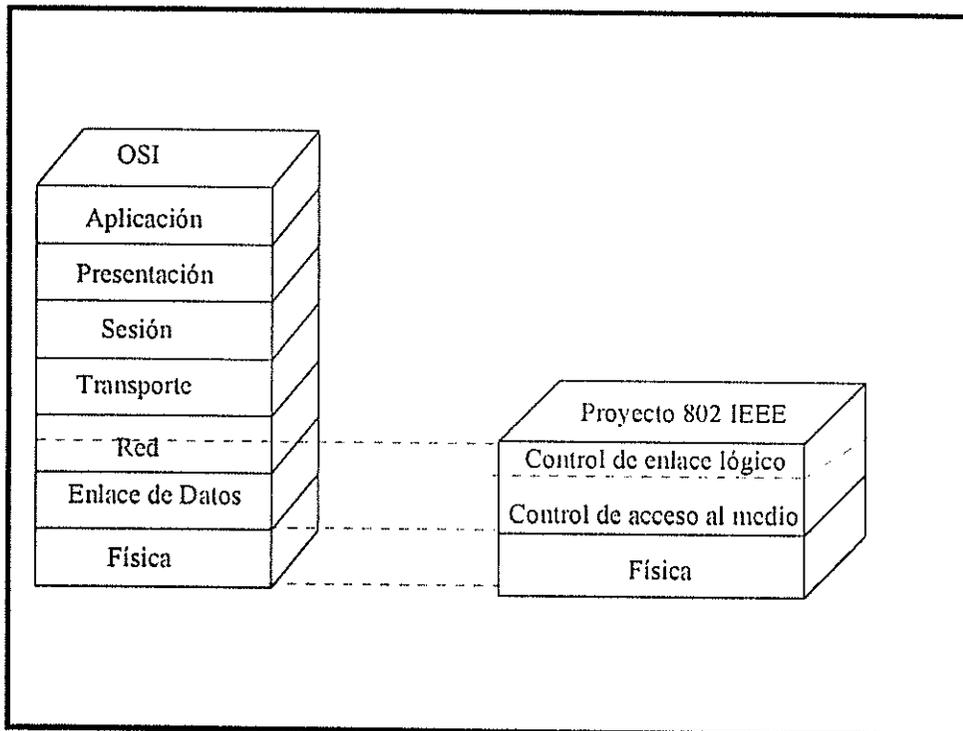


Fig. No. 4.13 OSI y el Proyecto IEEE 802.

IEEE 802.2 :CAPA DE CONTROL DE ENLACE LÓGICO (LLC)

La capa de control de enlace lógico LAN está descrita por el estándar IEEE 802.2. Este estándar es un elemento común en todos los estándares IEEE 802. Como la raíz de la jerarquía, IEEE 802.2 aísla las capas más altas de la arquitectura de una red de las específicas en una implementación LAN en particular. El estándar IEEE 802 define tanto las especificaciones de interface de servicio y de protocolos punto a punto. Es responsable del intercambio de las señales de control específico de protocolo entre las estaciones LAN. Entre otras de las funciones definidas por IEEE 802.2 están la organización del flujo de datos interpretación de comandos, generación de respuesta, control de error y recuperación.

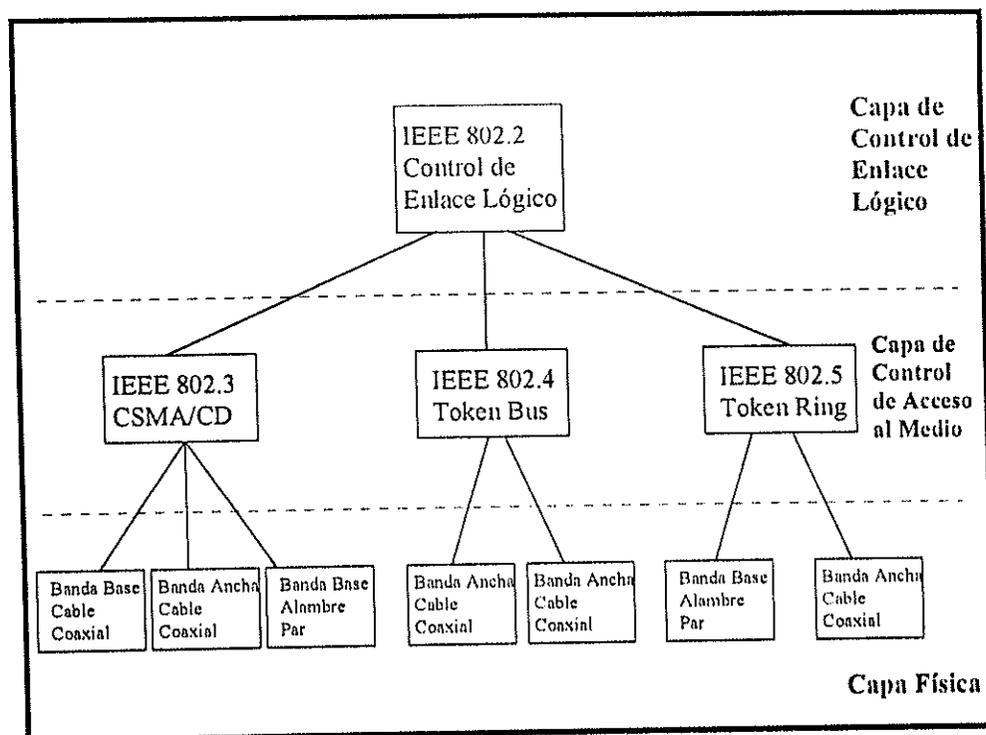


Fig. No. 4.14 Capas del Proyecto IEEE 802.

CAPA DE CONTROL DE ACCESO A LOS MEDIOS E IEEE 802.3

Las capas de control de acceso a los medios (Media Access Control o MAC), de los estándares IEEE 802, están diseñadas para soportar múltiples dispositivos que compiten por accesos, a un medio de transmisión físico individual. Las capas MAC definen las reglas que las estaciones deben de obedecer a fin de compartir un medio de transmisión común. El estándar de acceso a los medios contiene las siguientes funciones :

- Las funciones de la administración de acceso a los medios son utilizadas para controlar el compartimiento de un medio de transmisión entre todas las estaciones en una red de trabajo.
- Una función de enmarcar (framing) identifica el inicio y el final de un mensaje añadiendo información de encabezado e información final. La enmarcación (framing) facilita la sincronización emisor-receptor, la ruta y detección de error.
- Una función de direccionamiento identifica los dispositivos participantes en la emisión y recepción de mensajes, y determina las direcciones apropiadas de la red de trabajo.
- La función de detección de error asegura que el mensaje ha sido transmitido y recibido correctamente.

El proyecto 802 IEEE identificó el acceso múltiple por detección de portador con detección de colisión (CSMA/CD), métodos de token bus y de accesos de token ring, para soportar estándares MAC. Estos tres métodos de control de acceso descritos anteriormente, se interfazan con el mismo estándar de control de enlace lógico IEEE 802.2. El proyecto 802 ha nombrado sus métodos de control de acceso

elegidos como IEEE 802.3, IEEE 802.4, e IEEE 802.5 respectivamente. El primer método MAC, CSMA/CD, es mejor conocido por su utilización en las redes de trabajo Ethernet.

ETHERNET

Las redes de trabajo Ethernet originadas del desarrollo LAN, fueron puesta en funcionamiento por la corporación Xerox en su centro de investigación Palo Alto (PARC). El diseño resultó muy exitoso, y las especificaciones Ethernet definidas juntamente por Digital Equipment Corporation, Intel Corporation, y Xerox Corporation proporcionaron una contribución sustancial al estándar IEEE 802.3. Muchos vendedores de hoy ofrecen productos de redes Ethernet que son compatibles con varios sistemas operativos y el estándar IEEE 802.3. Por ejemplo, ambos Ethernet e IEEE 802.3 ofrecen encapsulación/descapsulación de datos, y encriptamiento/desencriptamiento de datos. La administración del acceso a los medios de IEEE 802.3 y las funciones de conexión al medio físico corresponden a la administración del enlace y la administración al canal del acceso de Ethernet.

Ethernet está diseñado para proporcionar simplicidad de la red de trabajo, bajo costo, flexibilidad y compatibilidad entre diferentes implementaciones Ethernet, alta velocidad, bajo retardo, estabilidad, y mantenimiento de las LANs basadas en Ethernet. Usualmente, Ethernet soporta transmisión de banda base, a una tasa de información de 10 Mbits/Seg., con un cable de longitud máxima de 500 metros.

IEEE 802.4: ESTÁNDAR TOKEN BUS

El estándar IEEE 802.4 es uno de los dos estándares basados en traslado de tokens IEEE 802. IEEE 802.4 define el estándar de traslado de token para implementaciones LAN sobre una topología de bus física. En el método de control de token bus IEEE 802.4, cada estación conoce la dirección de la estación desde la cual el token es recibido (predecesor), y la dirección de la estación a la cual debe transmitir un token (sucesor). Un token IEEE 802.4 es pasado (transmitido) de una estación a otra. Cuando el token es recibido la estación es autorizada para transmitir mensaje por una cantidad de tiempo predefinida. Cuando el límite de tiempo ha transcurrido o cuando la estación no tiene mensajes que transmitir, la estación transmisora envía el token a la siguiente estación. Aunque la topología física de la red IEEE 802.3 toma la forma de un bus, la operación lógica se encuentra en la forma de anillo lógico. El estándar IEEE 802.4 soporta tasas de información de 1 a 10 Mbits/seg.. El protocolo de automatización de manufactura de la General Motors es un ejemplo de la arquitectura de red basado en el estándar IEEE 802.4.

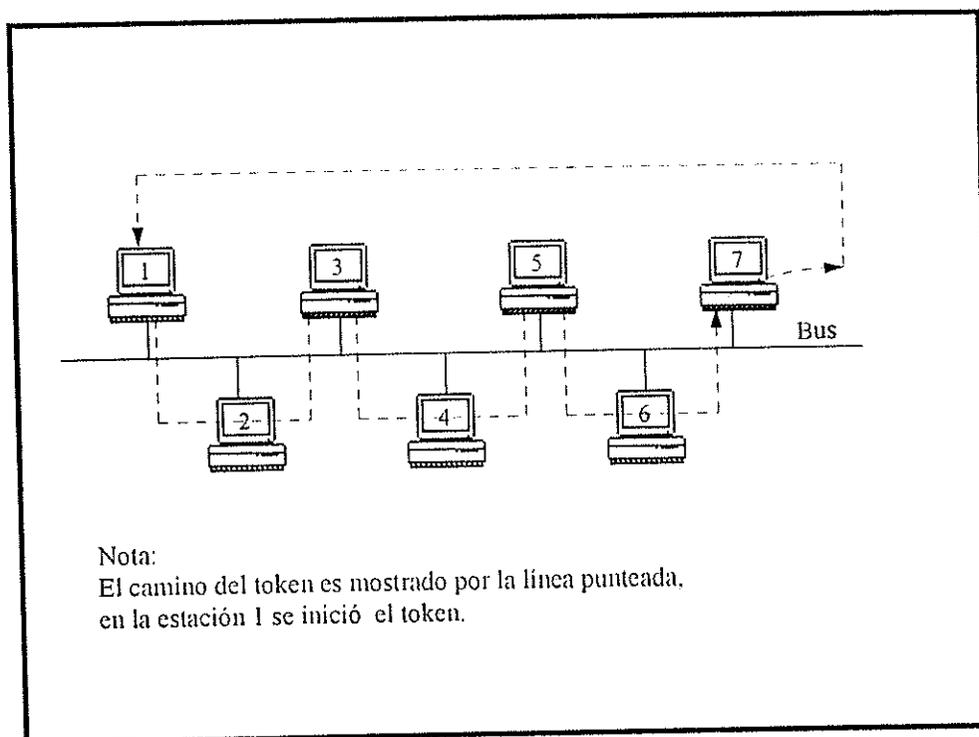


Fig. 4.15 Traslado de token bus IEEE 802.4

IEEE 802.5 : ESTÁNDAR TOKEN RING

El estándar token ring IEEE 802.5 utiliza un modelo lógico similar a la arquitectura de token bus IEEE 802.4.

La arquitectura token ring IEEE 802.5 consiste en un anillo lógico implementado que utiliza topología de anillo físico. La transmisión de tokens y mensajes fluye en una dirección individual. Las unidades de información son trasladadas de una estación a la siguiente, y cada estación actual como una repetidora.

Similarmente al IEEE 802.4 el derecho de transmitir en una red de token ring es controlada por un token. Se dice que un token "libre" es el que permite transmitir a la estación. Cuando una de las estaciones que necesita enviar mensajes recibe un token libre, cambia el token a un estado "ocupado" y es autorizado a transmitir mensajes por un espacio de tiempo predefinido. El token ocupado está incluido por una estación transmisora como parte de un mensaje. Cuando el límite de tiempo ha transcurrido, la estación transmisora envía el token y un mensaje a la siguiente estación. Cada estación que recibe el mensaje, revisa la dirección del mensaje para verificar si debe procesar el mensaje. Conforme el mensaje viaja alrededor del anillo, el transmisor del mismo finalmente lo recibe, cambia el token a un token libre y remueve el mensaje, además de las direcciones de estación, los tokens pueden contener prioridades de mensaje. El estándar IEEE 802.5 especifica una técnica de transmisión de banda base, sobre un par de alambres trenzado con tasa de información de hasta 4 Mbits/Seg., y sobre un cable coaxial con tasas de información de hasta 40 Mbist.Seg. El IEEE 802.5 es la base para las ampliamente usadas redes de trabajo de área local de token ring de IBM.

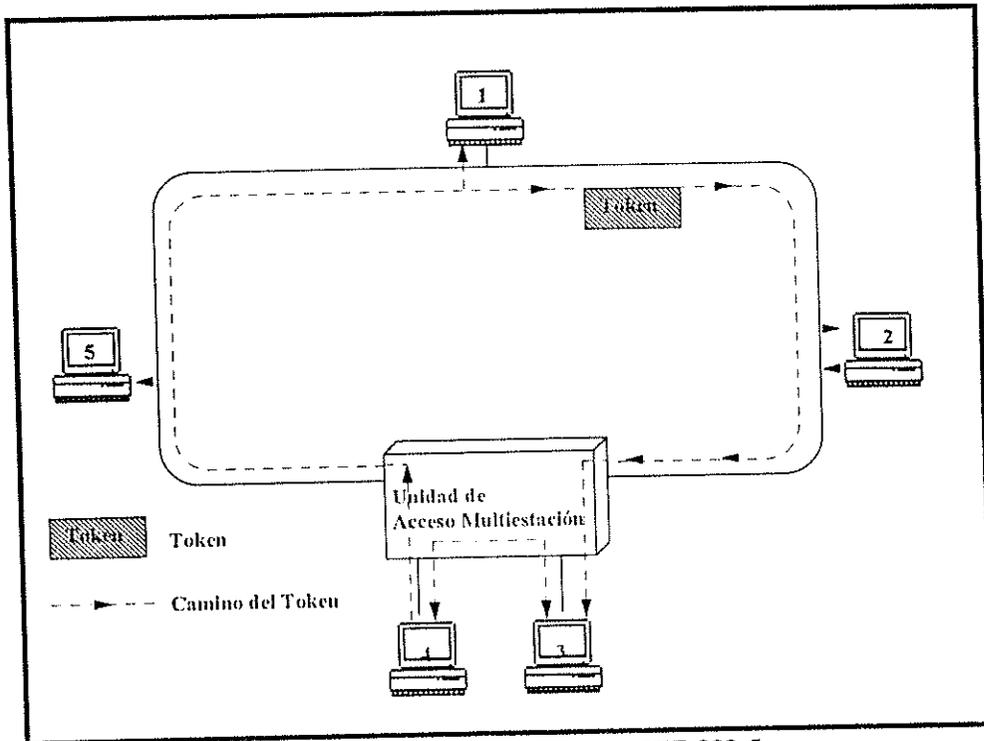


Fig. No. 4.16 Traslado de token ring IEEE 802.5

CAPITULO 5

PROTOCOLOS UTILIZADOS EN LANS

5.1 NetBIOS/NetBEUI

El sistema de entrada/salida básico de red de trabajo (Network Basic Input Output System NetBIOS) y los protocolos de Interface de usuario extendido NetBIOS (NetBIOS Extended User Interface NetBEUI), fueron diseñados por IBM y Microsoft para soportar comunicación de red en el ambiente de red de área local (LAN) de pequeño a mediano tamaño. El ambiente completo del protocolo es ilustrado en la figura No. 5.1 y explicado a continuación.

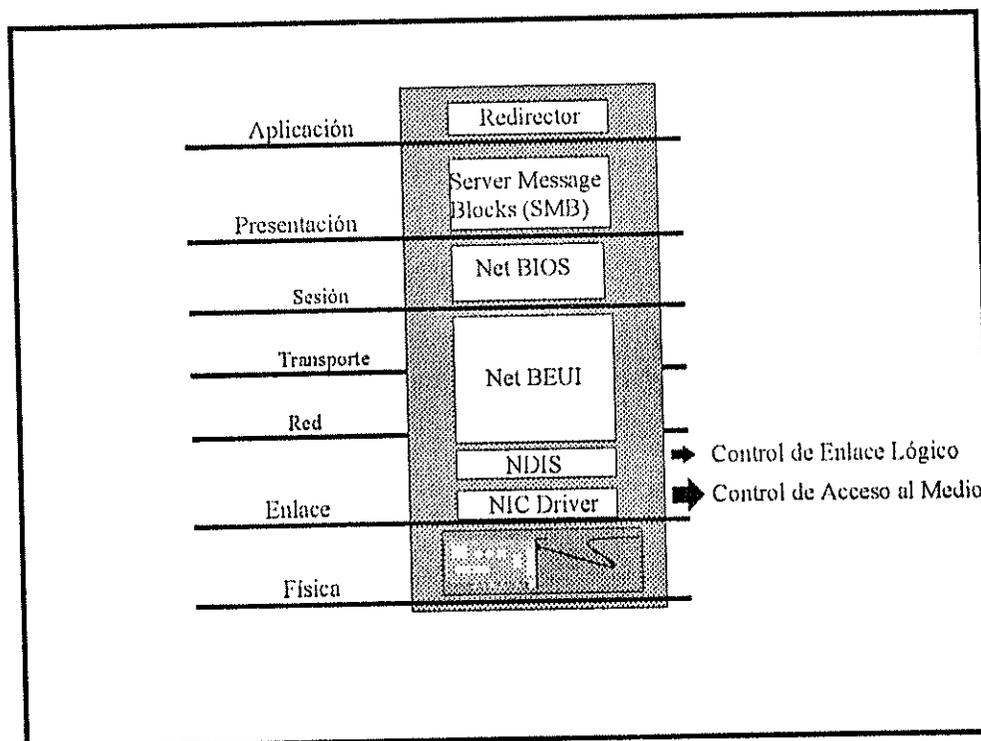


Fig. No. 5.1 Ambiente del protocolo NetBIOS/ NETBEUI.

- Redirector. Dirige los requerimientos de la red o a los servidores de red y los comandos locales al sistema operativo local.
- Bloques de mensajes de servidores (Server Message Blocks o SMB) proporciona el lenguaje punto a punto y los formatos requeridos para que las computadoras puedan comunicarse con otras.
- NetBIOS. Un protocolo de la capa de sesion en relación con el modelo de protocolo OSI. Configura las sesiones de comunicación entre computadoras y mantiene esas conexiones.
- NetBEUI. Proporciona los servicios de transporte de la información que servirá de base.

- Especificaciones de interface del controlador de la red de trabajo (Network Driver Interface Specification o NDIS). Una especificación recientemente desarrollada por Microsoft que proporciona el camino para soportar otros protocolos tales como TCP/IP en una tarjeta de interface individual.

NetBIOS y NETBEUI están implementados en un determinado número de sistemas operativos y en sistemas operativos de red de trabajo incluyendo OS/2, Windows para trabajo en grupo, Windows NT, Administrador LAN de Microsoft y servidor LAN de IBM. Microsoft también soporta el Protocolo de Control de Transmisión/Protocolo Internet (TCP/IP); ha sido colocado en todos sus productos de red de trabajo actuales y futuros. Una razón para ello es que NetBIOS no es un protocolo ruteable y no es conveniente en el ambiente de red de trabajo de área amplia (WAN).

5.1.1 NetBEUI

NetBEUI fue desarrollado por IBM en 1985 como un protocolo de transporte para LANs de tamaño pequeño y mediano. NetBEUI es patrocinado por Microsoft en sus productos de red de trabajo, que incluye windows para trabajo en grupo y windows NT. Las redes de trabajo de IBM y Microsoft pueden comunicarse gracias a la implementación NetBEUI y NETBIOS.

NetBEUI es un protocolo de capa de transporte y red en relación con el modelo de protocolo OSI. Este se integra con el NetBIOS para proporcionar un sistema de comunicación en el ambiente LAN de trabajo en grupo. NetBEUI proporciona los servicios de transporte de información requeridos por NETBIOS. Piense en una llamada telefónica, NetBIOS sería la persona efectuando la llamada y NetBEUI es similar al programa de control que trabaja con el sistema conmutador que sirve de base para completar la llamada.

5.1.2 NETBIOS

NetBIOS fue diseñado con la premisa de que las computadoras personales en un LAN solamente necesitan hablar a otras PCs sobre el mismo LAN. Este es un protocolo de capa de sesión con relación al modelo de protocolo OSI, y fue originalmente desarrollado por SyteK Inc. para que IBM lo utilizara en sus redes de trabajo de banda ancha.

NetBIOS es una interface de programa de aplicación (API) que los programadores utilizan para crear aplicaciones LAN para ambientes: servidores LAN IBM, administración LAN Microsoft y OS/2. Named Pipes¹⁰ es una extensión de red de trabajo para OS/2, que proporciona características similares pero más avanzadas. NetBIOS y named pipes existen en los ambiente LAN, como los protocolos en los cuales varias aplicaciones son construidas.

NetBIOS proporciona los siguientes servicios :

- NetBIOS establece nombres únicos lógicos para nodos en la red simplificando la tarea de referirse a otros sistemas. El nombre contiene de 1 a 15 caracteres de largo y específicamente identifica una estación de

¹⁰ Tuberías con nombre (named pipes) : Interface del alto nivel para trasladar información entre procesos que corren en computadoras separadas, las cuales están conectadas por la red.

trabajo tanto para la computadora como para el usuario. Este nombre es especificado cuando un computador de la red por primera vez arranca o se conecta a la red de trabajo. Los mensajes pueden ser enviados a grupos de nombre.

- NetBIOS establece una sesión de conexión orientada sobre las cuales los nodos conversan unos con otros. La sesión tiene lugar en una conexión lógica o circuito. NetBIOS configura, mantiene y desconecta una sesión. Las estaciones de trabajo se comunican en tiempo real con una entrega garantizada de información a través de las confirmaciones de entrega de mensajes.
- NetBIOS también puede proveer servicios datagrama sin conexión en los cuales los mensajes son dirigidos a otro sistema sin ningún arreglo previo de conexión o monitoreo de flujo de paquetes.
- Las aplicaciones de red, que corren sobre redes NetBIOS confiable, usan NetBIOS para localizar otros recursos y establecer conexiones con esos recursos para el intercambio de información. Los recursos pueden ser otras aplicaciones.
- NetBIOS emite información hacia toda la red (broadcasts) acerca de la localización de los servidores y sus nombres. Esta emisión broadcast puede sobrecargar la red con un exceso de paquetes, y causar problemas entre las redes. Filtrando la información en puentes y ruteadores, puede resolver este problema.
- NetBIOS no es ruteable, en parte por su sistema de nombramiento de 15 caracteres. NetBIOS debe encapsularse en paquetes de otros protocolos para enviarse sobre las inter-redes. El protocolo TCP/IP tiene procedimientos de encapsulación NetBIOS. Novell soporta NetBIOS en sus redes IPX, y DEC soporta NetBIOS sobre las redes DEC.

Los servidores usan las emisiones hacia toda la red (broadcasting) características de NetBIOS para localizar las estaciones de trabajo y los dispositivos en sus dominios (Zona de administración). Si una estación de trabajo responde a los mensajes, el servidor ya no envía más mensajes. Si una estación de trabajo no responde porque está apagada o porque tiene problemas en la tarjeta de interface de red, el servidor envía mensajes a cada cierto tiempo, en un intento de obtener respuesta. A causa de que los mensajes son enviados a cualquier lugar en la red, incluyendo enlaces con redes WAN, éstos llegan a encarecer en términos de uso de ancho de banda y costo de enlace WAN. Una función de Directorio de Nombre Remoto (RND) puede ser usado para convertir estas emisiones hacia toda la red en emisiones datagrama no globales para que no llegen a todos los enlaces WAN.

NetBIOS está bien establecido y es soportado en muchos ambientes, aun en los nuevos protocolos como named pipes. NetBIOS es asimismo bien comprendido y existen muchas aplicaciones que lo utilizan.

5.2 ESPECIFICACION DE INTERFACE CONTROLADOR DE RED (NETWORK DRIVER INTERFACE SPECIFICATION o NDIS)

La especificación de interface controlador de red (NDIS) de Microsoft fue diseñada para dar acceso a los usuarios de la red a una variedad de protocolos quitando los protocolos en la tarjeta de red. En el diseño, los protocolos no necesitan conocer nada acerca de la tarjeta de interface. Aquí no hay ninguna interface específico de tarjeta, solamente una interface común para los protocolos, como se muestra en la figura 5.2. Para usar una tarjeta NDIS, usted instala la tarjeta y su controlador, carga todos los protocolos que se desean usar, y los cuales son unidos con un comando llamado NETBIND.

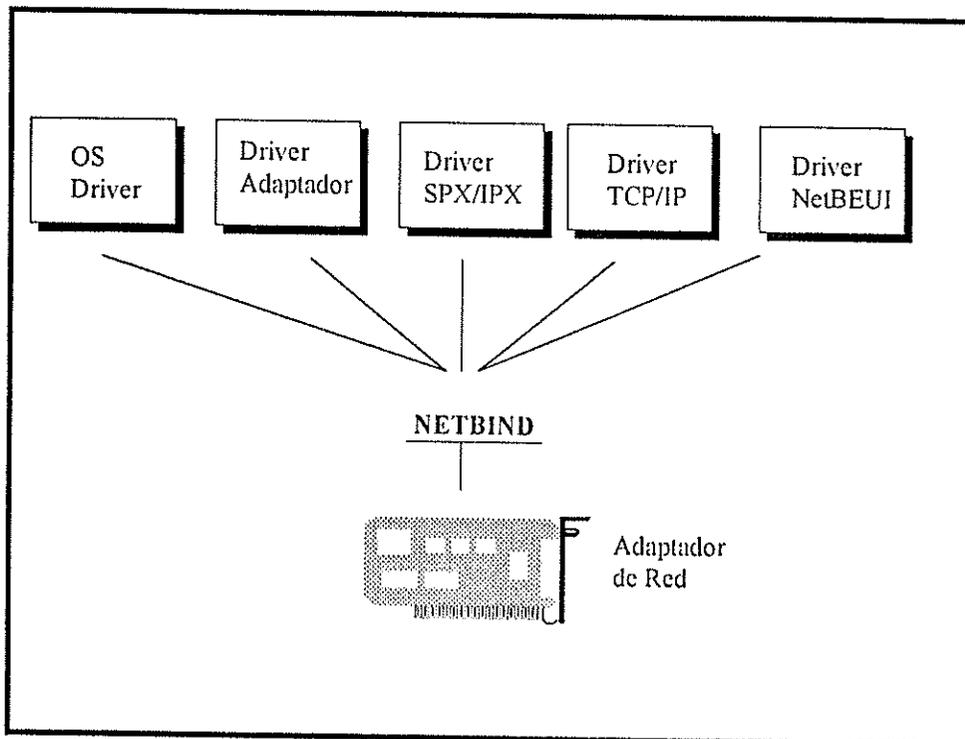


Fig. No. 5.2 Interface NDIS de Microsoft.

El propósito principal de NDIS es permitir la carga de pilas de protocolos múltiples en el servidor o en la estación de trabajo para que los usuarios puedan comunicarse con una variedad de protocolos. Antes del NDIS (o la comparable interface de enlace de datos abierto de Novell ODI), era necesario descargar una pila de protocolo y luego cargar otro; posiblemente apagando el computador, no era fácil la descarga de la pila de protocolos. NDIS permite múltiples direcciones de pilas de protocolos en una tarjeta de interface de red (network interface card o NIC) y los sistemas sobre la red, como se muestra en la figura 5.2. NDIS, permite a las estaciones de trabajo tener varias NICs alrededor de cuatro, y cada NIC puede soportar hasta cuatro pilas de protocolos.

Por ejemplo, un usuario que necesita interconectarse con ambos servidores, un servidor NetWare de Novell (Internet Packet Exchange o IPX) y un sistema basado en UNIX (Transmission Control Protocol/Internet Protocol o TCP/IP) conectados a la misma red, simplemente carga las pilas de protocolos IPX y TCP/IP juntamente. NDIS sirve de árbitro entre las dos pilas de protocolos y asegura que el ingreso de los mensajes del servidor NetWare sean dirigidos a la pila de IPX y los mensajes TCP/IP sean dirigidos a

la pila TCP/IP. NDIS provee cualquiera de los protocolos con acceso a la tarjeta de interface de red y al medio. Si el servidor NetWare y el sistema UNIX están en redes separadas, la estación de trabajo necesita dos tarjetas de interface y debe de cargar la pila de protocolo para cada uno, como se enseña en la figura No. 5.3.

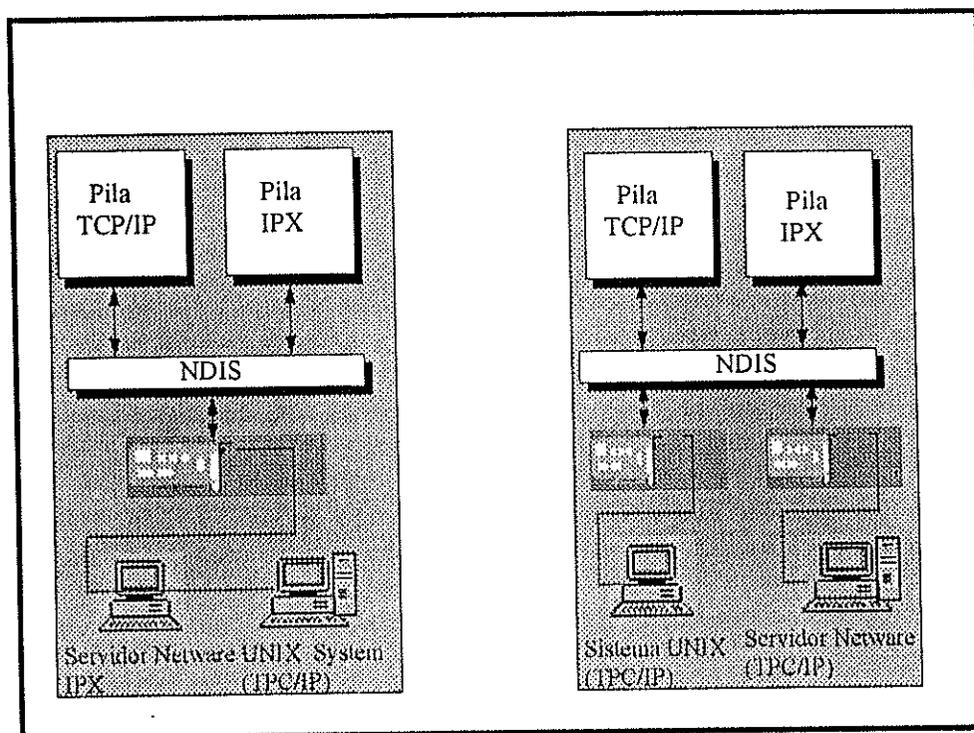


Fig. No. 5.3 Configuración NDIS.

De la misma manera, el servidor NetWare de Novell o el servidor avanzado Windows NT de Microsoft pueden correr múltiples pilas de protocolos para que los usuarios en diferentes tipos de estaciones de trabajo (DOS, UNIX, Macintosh, y otros) puedan acceder a los servidores.

Cuando NDIS es cargado durante la fase de inicialización (boot), lee los conectores o el archivo llamado `PROTOCOL.INI`. Este archivo especifica los NICs (la tarjeta de interface de red) instalada en la estación y la pila de protocolo es unida a cada tarjeta.

Para proporcionar soporte para NDIS en compañía de ODI, Novell desarrolló unas especificaciones de soporte de red de interface de enlace de datos abierto (Open Data-Link Interface Network support specification o ODINSUP). Este permite la coexistencia de las interfaces controladoras de red ODI y NDIS, por lo tanto las estaciones de trabajo pueden ser conectadas con sistemas ODI como servidores IPX y cualquier sistema NDIS, como un servidor de administración LAN NDIS.

5.3 TUBERÍAS (PIPES)

Tuberías (pipes) son comunicación interprocesos (IPC) característica de los sistemas operativos UNIX y OS/2. Son como claves en las cuales un proceso corriendo en el sistema operativo multitarea puede almacenar información que desea trasladarse asimismo a otro procesos corriendo en el computador, o múltiples procesos corriendo en el computador. La información es almacenada en una base de primero en entrar primero en salir (PEPS) y fluye como una corriente de bits que no es alterada durante la transmisión. Un pipe es abierto como un archivo y leído o escrito en la manera de un archivo. Son unidireccionales; un pipe es utilizado para leer y otro para escribir información. Los pipes existen en capa de presentación, relativo a modelo de protocolo de interconexión de sistemas abiertos (OSI).

Named pipes (tuberías con nombre) son estructuras lógicas en un servidor que otros sistemas accesan cuando necesitan utilizar los recursos del servidor. Named Pipes son una extensión de red del mecanismo IPC en OS/2 y de los productos LAN que corren en él, tales como el servidor IBM LAN y el administrador LAN de Microsoft. Un named pipe es un canal de comunicación utilizado para transferir comunicación entre programas, procesos y dispositivos sobre la red de trabajo. Esta es la base para el mecanismo de comunicación entre el cliente y aplicaciones avanzadas cliente/servidor tales como servidores SQL y servidores de comunicación.

La Figura 5.4 ilustra el proceso pipe. A la izquierda, dos procesos corriendo en el mismo computador comparten información utilizando un pipe; a la derecha, dos procesadores corriendo en procesadores separados comparten información usando un named pipe. Observe que los named pipes tiene un nombre asignados para diferenciarlos de otros pipes que podrian estar disponibles en la red de trabajo.

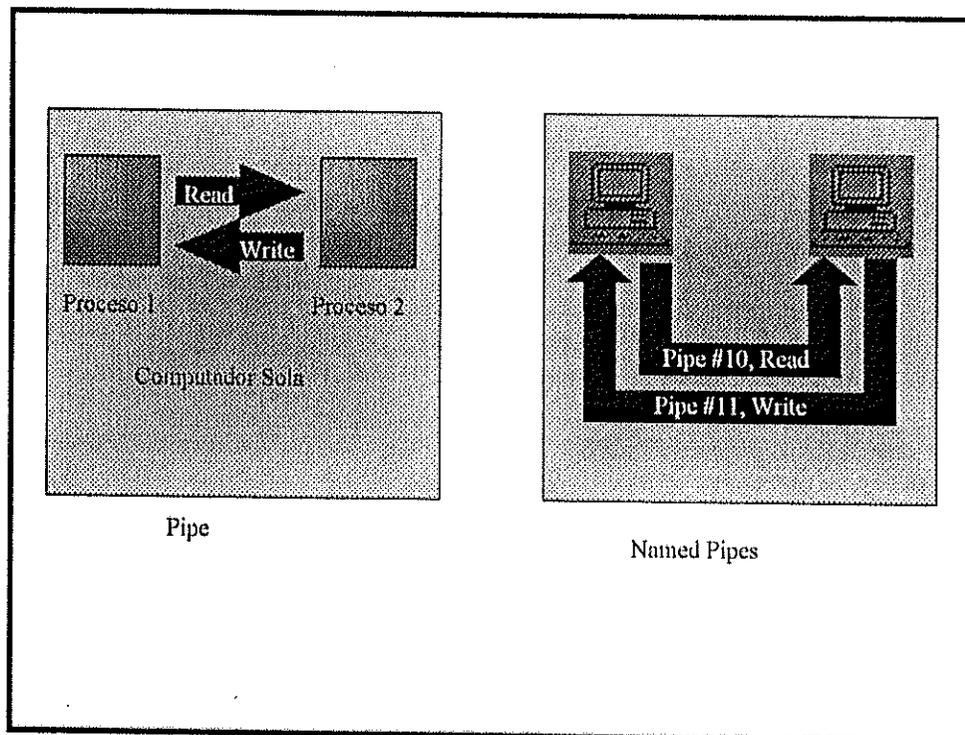


Fig. No. 5.4 Pipes y Named Pipes.

5.4 TUBERÍAS CON NOMBRE (NAMED PIPES)

Las tuberías con nombre (named pipes) es una interface de alto nivel para pasar datos entre procesos que están corriendo en computadoras separadas, las cuales están conectadas por la red. Named pipes es una extensión del sistema operativo OS/2, y está implementado en el Administrador LAN de Microsoft y los sistemas operativos de redes de servidores LAN IBM. Esta es la versión de red de facilidades de comunicación interprocesos (interprocess communication o IPC), la cual provee un interface entre procesos en sistemas simple y multitarea.

Named pipes es parte del grupo de facilidades de comunicación de interprocesos en el ambiente OS/2 (y algunos otros ambientes) que son usados para procesos corriendo concurrentemente, que necesitan comunicarse con otro. Esto incluye :

- **MEMORIA COMPARTIDA:** es un bloque de memoria compartida que dos programas usan para leer y escribir información común. Un conjunto de reglas de acceso que prevengan a los programas de escribir la misma data simultáneamente. La memoria es usada para mantener la información e intercambiar esta entre programas.
- **SEMAFOROS:** es un semáforo es como una bandera o señal que previene a los procesos corriendo separadamente, de accesos entrada/salida de los puertos simultáneamente y otros recursos computacionales.
- **MENSAJERÍA:** es el proceso de mensajería de almacenamiento y reenvío (store-and-forward), en el cual un procesos intercambia mensajes con otro proceso. El mensaje puede contener un requerimiento para servicio o data. La mensajería es manejada en base PEPS (primero en entrar, primero en salir). En la red, la mensajería puede no ser apropiada para aplicaciones que requieren respuesta inmediata, ya que puede tomar algún tiempo antes de que el mensaje sea realmente entregado, pero la mensajería es útil para algunos tipos de consultas de bases de datos e inicialización de procesos.
- **PIPES:** es una forma de comunicación interprocesos similar a las llamadas de procedimiento remoto que toma lugar entre dos procesos en la misma máquina.

Comparado contra un pipe, named pipes es un proceso de comunicación entre dos sistemas de computadoras remotas que están conectadas en la red. Una conexión lógica es configurada entre dos procesos remotos y la transferencia de información toma lugar sobre la red física. El aspecto más importante de named pipes es que están habilitados para las aplicaciones cliente/servidor en ambientes de computación distribuidos. Un proceso front-end¹¹ en la máquina del cliente puede interactuar con un proceso back-end¹² en la máquina servidora. Por ejemplo, un cliente puede hacer requerimientos de archivos a un servidor de archivos sobre la red, o una consulta a una base de datos en un servidor de base de datos remoto.

La interface named piped es mejor que NetBIOS, para el manejo de transacciones en la red y puede hacer esto con una sola llamada comparado con las muchas operaciones requeridas para efectuar las mismas acciones a través de NetBIOS. Sin embargo, la interface named pipes no proporciona las características de

¹¹ Proceso front-end : proceso que interactua con el usuario, que es visible para el usuario.

¹² Proceso back-end : proceso que presta servicios y no es visible a los usuarios.

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
BIBLIOTECA

NetBIOS, tales como, servicios datagrama sin conexión y las funciones de nombramiento que permiten que los mensajes sean enviados a los grupos.

5.5 INTERCAMBIO DE PAQUETES EN REDES INTERCONECTADAS (INTERNETWORK PACKET EXCHANGE o IPX)

IPX es protocolo de red, construido punto a punto para NetWare de Novell; éste se originó del protocolo de sistema de trabajo Xerox XNS, el cual fue desarrollado en la década de los 70s. NetWare es un sistema operativo cliente/servidor que proporciona a los clientes servicio de archivo compartido, servicio de impresión, seguridad y una variedad de otras características opcionales, tales como bases de datos, servicios de aplicaciones, comunicación, fax y correo electrónico. IPX es el medio primario que proporciona estos servicios.

5.5.1 PAQUETES IPX

Un paquete es la colección de bits que contiene datos y un encabezado de información. La información del encabezado incluye: la dirección fuente y destino, así como un control de información para manejar los errores y mantener los paquetes en orden. Los paquetes son enviados a su destino como una corriente de bits enmarcados a través de un canal de comunicación. Cada paquete es un bloque separado de información, que potencialmente tienen diferentes direcciones de otros paquetes que atraviesan la red. La información en los paquetes puede ser un requerimiento de un servicio de un cliente o una respuesta de un servidor. La información puede ser información texto que es transferida de un sistema a otro. Un paquete típico, puede estar comprendido en un tamaño entre 512 y 6,500 bits y puede tomar muchos paquetes para transferir un archivo grande sobre la red.

IPX es un protocolo datagrama que opera en la capa de red del modelo de protocolo OSI. Datagrama es otro nombre que se le da a los paquetes que son transferidos con comunicación sin conexión. Sin conexión implica que los paquetes de información son enviados de la fuente al destino sin que antes haya habido establecimiento de la conexión y sin un monitoreo mutuo de intercambio de paquete por ambos. La fuente simplemente asume que el destinatario puede manejar cualquier tipo de información que éste envía. El destino recibe los paquetes y los reordena. Si algún paquete se pierde, éste manda un requerimiento de transmisión del paquete.

En contraste, el protocolo hermano de IPX el SPX o intercambio de paquetes en secuencia (Sequenced Packet Exchange) es una conexión orientada y reside en la capa de transporte del modelo de protocolo OSI. Conexión orientada significa que una sesión es configurada entre dos sistemas de comunicación para monitorear el flujo de paquetes de datos. Mientras que la sesión de conexión orientada SPX requiere más tiempo para ser configurada, ésta puede garantizar la entrega apropiada de la información en sentido y tiempo. Sus paquetes son también de un tamaño menor debido a que la información de dirección completa no es requerida, por otro lado una sesión sin conexión es rápida y eficiente, pero sus paquetes son más grandes en tamaño, ya que requieren dirección completa para llevarlos a su destino.

En la mayoría de los casos, los métodos sin conexión son preferidos sobre las LANs relativamente libres de error debido a que con el tiempo son más eficientes en la entrega del tipo de datos que comúnmente son intercambiados en un ambiente de red de trabajo. Esta información generalmente llega incompleta y requiere demasiada sobrecarga para cada fallo. En contraste, las sesiones orientadas a la comunicación son

preferidas cuando se usan utilidades tales como, la consola remota de NetWare. Esta utiliza SPX para configurar una conexión relativamente permanente, entre una estación de trabajo de un administrador y un servidor para los propósitos de proporcionar información continua y en línea acerca del estado de un servidor. En vista de que esta conexión permanece en uso y porque los paquetes son transferidos sobre una base regular, se prefiere el SPX orientado a la conexión.

5.5.2 USANDO IPX

IPX está contruido en el sistema operativo NetWare; de esta manera, las estaciones de trabajo solamente necesiten instalar el protocolo de comunicación y correr un paquete de software de cliente para hablar con los servidores NetWare. IPX no se utiliza en muchos ambientes que no sean NetWare a pesar de que se ha mostrado interés en usarlo para acceder a Internet.

El soporte de IPX está instalado en la estación de trabajo, basada en DOS, OS/2. Las versiones actuales de software incluyen Open Data-Link (ODI) de Novell, el cual proporciona el camino para cargar múltiples pilas de protocolos, tales como IPX y TCP/IP de ambos lados. El software solicitante es básicamente una utilidad de redirección que busca los comandos usados por el usuario y los envía al sistema operativo local o los envía al sistema operativo de la red sobre la red. Si el requerimiento está destinado para la red de trabajo, el software solicitante empaqueta el requerimiento en un paquete IPX y éste es ejecutado por la tarjeta de interface de la red, la cual envía el paquete como un flujo de bits. Actualmente, los paquetes pueden tener diferentes destinos:

- A un servidor sobre el mismo segmento LAN
- A una estación de trabajo sobre el mismo segmento LAN
- A un servidor o estación de trabajo en otro segmento LAN

El encabezado de la información en el paquete, contiene la dirección del segmento LAN. Cuando el paquete es una emisión para toda la red (broadcast), un ruteador deberá ver esta información y enviarla el paquete al segmento apropiado de la red, si ese fuera el segmento destino.

NetWare de Novell soporta una amplia variedad de enlaces de datos (data links) y transmisiones IPX sobre estos. Los enlaces soportados incluyen ARCNET, LocalTalk, Ethernet, Token Ring, FDDI y otros.

En el ambiente NetWare, los siguientes protocolos están íntimamente ligados con IPX :

NETWARE CORE PROTOCOL (NCP), NCP es protocolo principal para la transmisión de información entre un servidor NetWare y sus clientes. NCP maneja los requerimientos de logons y muchos otros tipos de requerimientos al sistema de archivos e impresoras. IPX es el protocolo base que acarrea la transmisión. NCP es un protocolo LAN, que originalmente fue diseñado asumiendo que el servidor y la estación de trabajo estarían relativamente cercanos. Cuando un ruteador está involucrado y las conexiones están hechas sobre enlaces de redes de área amplia (WAN), NCP causa congestión de tráfico. Este usa un esquema de requerimiento/respuesta para manejar las comunicaciones servidor/estación de trabajo. Si una

estación hace el requerimiento, éste debe de esperar la respuesta antes de hacer otro requerimiento. Si se conoce que un requerimiento causa exceso de tráfico, hay varios esquemas para resolver este problema.

SERVICE ADVERTISING PROTOCOL (SAP). Los servidores NetWare y los ruteadores sobre las redes NetWare usan SAP para enviar mensajes a toda la red cada 60 segundos, para indicar los tipos de servicio que estos pueden proporcionar.

ROUTING INFORMATION PROTOCOL (RIP). NetWare usa RIP para mantener informados a los ruteadores de la red, de los últimos cambios de los caminos de la ruta en la red. Los servidores NetWare pueden actuar como ruteadores. RIP usa algoritmos de distancia-vector (DVAs) para calcular el camino de la ruta. En DVA, las decisiones están basadas en el menor número de saltos para un destino. Los RIPs de Novell incluyen un campo que indica la velocidad de un enlace, la cual proporciona otra forma para la definición de caminos para que los paquetes atraviesen la red y lleguen a sus destinos.

Las tablas de ruteo RIP son intercambiadas con otros ruteadores, cada 60 segundos, y los ruteadores reconstruyen sus tablas de rutas basados en la nueva información. Algunos problemas pueden ocurrir si un ruteador llegara a quedarse retrasado en la reconstrucción de la tabla de ruta, la cual puede ocurrir si un ruteador está conectado sobre un enlace lento sobre una red WAN. Adicionalmente, el intercambio de la tabla de ruta puede agregar bastante sobrecarga a la red LAN, la cual puede causar más congestionamiento, y retrasar las actualizaciones de la tabla de ruteo. Si una ruta falla, el retraso requerido para reconstruir las tablas de ruteo pueden retrasarse tan rápidamente como una nueva ruta es establecida.

Estos problemas han empezado a resolverse con el nuevo protocolo de ruteo de Novell, llamado NetWare Link Services Protocol (NLSP).

5.6 TCP/IP (TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL)

Mientras el SNA de IBM fue desarrollado primordialmente para satisfacer los requerimiento de red de trabajo y comunicación de los usuarios del sistema IBM. Se han hecho significativos esfuerzos de investigación para desarrollar tecnologías que permitan a las redes de trabajo diferentes, físicamente heterogéneas para lograr interconectarse. La mayor parte la investigación fue patrocinada por varias agencias del gobierno de los E.U.A..

Estas organizaciones debieron darse cuenta y apreciar la importancia y el potencial de la interconexión de la red, algunas veces llamadas internetworking o internet. La tecnología líder Internet es el resultado de investigaciones patrocinadas por la Agencia de Proyectos de Investigación Avanzada de la Defensa (DARPA). Esta tecnología conocida como la transmisión de control protocolo/Internet Protocolo (TCP/IP), incluye un conjunto de estándares de red de trabajo que especifican los detalles de las comunicaciones de computador, así como también las convenciones para interconexión de redes de trabajo y ruteo de redes.

5.6.1 TCP/IP Internetworking (Interconexión de redes con TCP/IP)

TCP/IP constituye la tecnología base para una gran Internet, que conecta establecimientos de mayor investigación, universidades, y agencias del gobierno. El resultado es Internet, que conectan entidades bien conocidas como: NSF (The National Science Foundation), DOE (The Department of Energy), DOD (The Department of Defense), NASA (The National Aeronautics and Space Administration), y otros.

Las características de TPC/IP difieren de otros protocolos de red de trabajo en los siguientes aspectos :

- Independencia tecnológica de red de trabajo. Por su diseño, TPC/IP es independiente de cualquier hardware de un proveedor en particular. TCP/IP define cómo las unidades de transmisión de información (datagrama) son transmitidas sobre una red de trabajo particular.
- Interconexión de cualquiera a cualquiera. Esto permite que cualquier par de sistemas de cómputo, se conecten por sí mismos, y pueden comunicarse asignando una dirección única, reconocible a través de Internet a cada sistema.
- Reconocimiento fuente-destino. Protocolos TCP/IP soportan reconocimiento de extremo a extremo, entre la fuente y el último destino del mensaje, aun si la fuente, y el destino residen en redes de trabajo físicamente separadas.

5.6.2 PROTOCOLOS TCP/IP

El TCP/IP es una arquitectura de comunicación de cuatro capas, que puede describirse conceptualmente como una jerarquía, construida en una interface de red de trabajo física, que proporciona interfaces de comunicación al hardware de la red de trabajo, ver figura 5.5.

La capa que está encima de los servicios de la red de trabajo es referida como la capa de protocolo de Internet (IP). Es un servicio sin conexión que trata acerca de la entrega de paquetes de información. (Es un servicio sin conexión debido a que trata a cada paquete independientemente de todos los demás). El IP es esencialmente un servicio datagrama para satisfacer lo siguiente:

- Definición de una unidad básica de transmisión de datos utilizada a través de la red de trabajo TCP/IP.
- Los paquetes de información de ruteo, que incluye el ruteo entre arquitecturas de redes de trabajo no similares (Por ejemplo, IEEE 802.LAN y X.25 WAN).
- El mejor esfuerzo en la entrega de la información (pero no garantizada).

El protocolo IP incluye reglas de cómo los paquetes deberían de ser procesados, cómo y cuando se deberían de generar mensajes de error, con base en estos los paquetes deberían de ser descartados. Sin embargo, el IP por su diseño, no garantiza entrega de información.

La entrega garantizada, concurrencia de la información y secuencia, son las responsabilidades de la capa de protocolo de control de transmisión (TCP). Adicionalmente, TCP se hace cargo del chequeo de errores, retransmisión, y conexión a aplicaciones en otros sistemas.

TCP es un protocolo de conexión orientado, que utiliza conexiones entre dos puntos, no puntos finales individuales, como en su concepto fundamental.

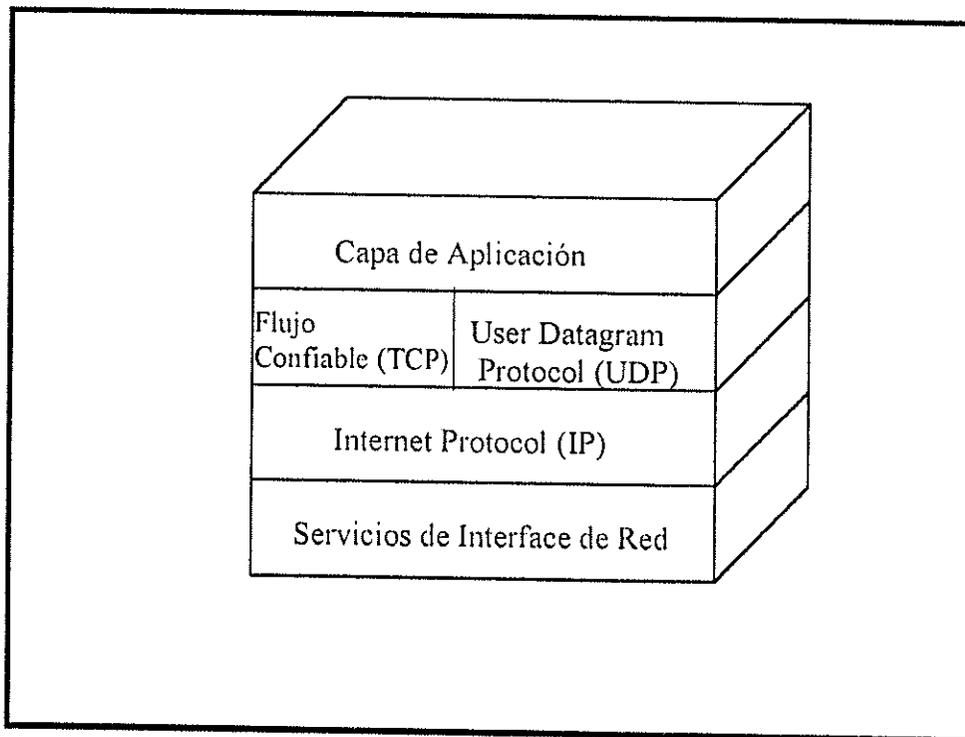


Fig. No. 5.5 Arquitectura TCP/IP.

5.6.3 DIRECCIONAMIENTO, CONECTORES (SOCKETS) Y SECUENCIAMIENTO

TCP/IP fue construido primariamente permitiendo a un programa particular en un nodo en particular en una red en particular conectarse a un programa residente remoto en un nodo de la red, que puede estar o no en la misma red; la red en cuestión puede ser de diferentes vendedores.

Para solventar el problema de interconexión, TCP/IP usa los siguientes mecanismos de direccionamiento que permite la entrega de los paquetes de información a través de distintas arquitecturas de redes :

- Cada programa (conocido como un proceso en TCP) es asignado a una dirección única en cada sistema, que es una dirección local.
- Esta dirección local única es combinada con la dirección particular del nodo para formar una dirección puerto.

Los procesos por ellos mismos no son una conexión TCP de punto final. TCP define los puntos finales, como una pareja (host,puerto), donde un host es la dirección IP para el host, y un puerto es un puerto TCP en este host.

Otro componente importante de una red TCP/IP es un conector. Los conectores son la base para las entradas/salidas de la red, construidos en los ambientes UNIX basados sobre el Software Berkeley de sistemas operativos distribuidos, que puede ser visto como el resultado de la combinación de dirección puerto con la dirección de la red local.

Una red típica TCP/IP puede contener un gran número de conectores, pero cada conector identifica únicamente a una aplicación específica en un nodo específico en una red específica. A través del mecanismo de conectores (socket), el IP puede obtener paquetes de los propios nodos, donde el TCP entregará los paquetes al propio programa (proceso) en ese nodo.

Dos conectores, que desean conectarse entre sí, deben utilizar un mecanismo particular para establecer tal conexión. TCP, siendo un protocolo de conexión orientada, proporciona varios de estos mecanismos. Uno de muchos caminos comunes para establecer una conexión TCP es usando una red abierta activa o pasiva. Cuando un conector (socket) que desea recibir información, se declara abierto y disponible para la llegada de tráfico, es un abierto pasivo. Un abierto pasivo puede ser hasta especificado completamente; esto significa que el conector (socket) que emitió un abierto pasivo le dice a la red de trabajo cuál conector puede conectarle.

A diferencia del abierto pasivo, el conector abierto activo intenta encontrar un conector (socket) con el que desea conectarse. Un abierto activo puede ser exitoso, sólo si el conector requerido coopera siendo un abierto pasivo o un abierto activo. Para establecer una conexión que soporte una sincronización correcta entre dos puntos extremos, TCP lo hace de tres maneras :

- El requiriente de la conexión envía una señal sincronizada y un número de secuencia hacia el punto de destino final.
- El receptor recibe la señal sincronizada (segmento), y envía de vuelta la confirmación, un número de secuencia y una señal sincronizada.
- Al recibir ambas señales, el requiriente envía la confirmación de vuelta al receptor.

Debido a que TCP es construido sobre un servicio de entrega de paquetes no confiable (IP), los mensajes pueden perderse, retrasarse, duplicarse, o entregados fuera de orden. El camino de tres maneras es tan necesario como suficiente para la propia sincronización. Éste garantiza que ambos lados estén listos para transferir datos (y conocerlos), y ambos han acordado los números de secuencia inicial. TCP utiliza el método de checksum¹³ para la detección de errores, y además mejora la entrega garantizada del protocolo

¹³ Método checksum para detección de errores : método en el cual se utiliza un algoritmo de suma caracteres de la información a enviarse y el resultado de está lo envían como parte del encabezado, de la misma forma el que recibe el mensaje, realiza el mismo procedimiento y la suma debe de coincidir con la que se recibe en el encabezado.

utilizando la retransmisión de mensajes en eventos de time-out¹⁴. Una vez que la conexión está establecida, TCP proporciona transferencia de datos de un conector a otro. Los métodos más populares de transferencia de TCP son los siguientes :

- La transferencia de datos segmentados permite a TCP a enviar datos en segmentos a través de la red. Los tamaños de los segmentos pueden ser ajustados para proporcionar la mejor eficiencia.
- El modo empuja, fuerza a TCP a enviar todos los datos sin consideraciones de eficiencia de la red que ésta envuelve. Éste es usado cuando se requiere una transferencia inmediata de los datos.

5.6.4 SERVICIOS INTERNET

Desde el punto de vista del usuario, la interconexión de redes TCP/IP parece ser un conjunto de programas de aplicación capaz de comunicarse con otros en la red. Las siguientes son las aplicaciones más populares de TCP/IP Internet.

- Correo electrónico (Electronic mail). Este permite a los usuarios preparar memos y documentos y enviarlos a usuarios particulares y a un grupo de usuarios de la red. TCP/IP soporta el Protocolo de Transferencia de Correo Simple (Simple Mail Transfer Protocol o SMTP). SMTP proporciona el retorno de recibos, mensaje/correo hacia adelante y otras características útiles.
- Transferencia de archivos. TCP/IP soporta programas de transferencia de archivos, los cuales permiten a los usuarios enviar grandes archivos (imágenes o librerías completas) a través de la red. Las facilidades de transferencia de archivos son proporcionadas por un mecanismo conocido como el Protocolo de Transferencia de Archivos (File Transfer Protocol o FTP).

FTP permite transferencia de registros, transferencia de un bloque (grupo de registros), o transferencia de imágenes (independientemente del contenido del archivo). FTP puede soportar algunas conversiones de caracteres (por ejemplo ASCII a EBCDIC) antes de empezar a transferir.

- Acceso remoto a un sistema remoto (login remoto). Este permite al usuario ingresar a una computadora y luego conectarse a una computadora remota, estableciendo una sesión interactiva con ésta. Para un usuario, esto se parece a la terminal de usuario (teclado y pantalla) que está directamente conectada al sistema remoto. Un ejemplo de una aplicación de acceso remoto de TCP/IP es TELNET una facilidad de terminal virtual que permite a los usuarios conectarse a un sistema remoto, si bien la terminal del usuario es conectada físicamente al sistema remoto.

¹⁴ Eventos time-out : eventos fuera de tiempo, eventos en los cuales un proceso se queda esperando un determinado tiempo para recibir la respuesta de un requerimiento enviado previamente, si la respuesta no es recibida, notifica al usuario que no hubo respuesta a tiempo.

5.7 IMPLEMENTACIONES LAN

Los estándares IEEE 802 definen arquitecturas LAN que proporcionan un conjunto común de funciones de comunicación. Estos estándares han sido ampliamente adoptados por proveedores de productos LAN. La estandarización permite diferentes (heterogéneos) tipos de equipo para ser conectados a la misma red de trabajo e interoperar con éxito.

SOLUCIONES LAN DE IBM

Para alcanzar la metas de conectividad heterogénea, transmisión rápida de información, alta confiabilidad, disponibilidad, y servicio, IBM ha desarrollado una variedad de productos y servicios LAN que soportan varias arquitecturas e interfaces LAN. La implementación de IBM de estándares de control a los medios incluye la red de trabajo token ring de IBM (IEEE 802.5) y la red de trabajo PC de IBM, ambas con banda ancha y banda base (IEEE 802.3). Una interface común y un nivel de control de enlace lógico (IEEE 802.2) es ofrecido a través del programa de soporte LAN de IBM y el nivel más alto de interface de programación a la red de trabajo NET-BIOS. Adicionalmente, IBM proporciona puentes (gateways) que conectan LANs del mismo tipo (por ejemplo, el programa puente token ring de IBM) y puentes (gateways) que interconectan LANs heterogéneas, así como LANs con computadores Host Y WANs (programa de interconexión de red de trabajo anillo/PC de IBM, programa de emulación de PC 3270 de IBM, APPC/PC y APPC para OS/2 EE).

Red de trabajo token ring de IBM. Esta arquitectura LAN de IBM es consistente con el estándar IEEE 802.5. Utiliza transmisión de banda base y tasas de información de hasta 4 Mbit/Seg. sobre un sistema de cableado basado en alambre trenzado protegido o desprotegido. La lógica de control necesaria para implementar control de accesos a los medios son proporcionadas por las tarjetas adaptadoras a la red de trabajo token ring de IBM, instaladas en estaciones LAN basadas en PCs. Hasta 8 estaciones pueden ser conectadas a una unidad de acceso multiestación de red de trabajo, para formar una sub red de trabajo de alambrado estrella. Tales unidades de acceso pueden ser conectados en una red de trabajo de anillo capaz de soportar más de 8 estaciones.

NETBIOS. La interface de capa de control de enlace lógico, compatible con la capa de enlace de datos de modelo de referencia OSI, está proporcionado por el programa de soporte LAN de IBM. Al mismo tiempo, el programa de soporte LAN proporciona una interface de capa de sesión OSI de niveles más altos - NETBIOS. El protocolo de transmisión full-duplex NETBIOS soporta transmisión de datos confiable sobre circuitos virtuales y servicios datagrama sin conocerlo. Adicionalmente a los servicios datagrama, NETBIOS ejecuta control de servicio de nombre y funciones de servicio de sesión. Aunque NETBIOS no conforma un estándar publicado, representa un estándar LAN de facto. Varios vendedores de LAN ofrecen sistemas operativos de redes que emulan la interface NETBIOS.

INTERFACES APPC. Una alternativa de interface de capa de sesión para las redes de área local IBM, es proporcionado por APPC/PC. Esta interface está basada sobre los protocolos de comunicaciones de programa a programa avanzado (APPC) definidos por SNA IBM (Systems Network Architecture) Logical Unit Type 6.2 (LU6.2). APPC/LU6.2 está diseñado para propósitos generales de comunicaciones punto a punto entre programas. El protocolo APPC soporta la sincronización entre dos programas (procesos) que es necesaria para el procesamiento de transacciones distribuidas e integridad de datos en ambientes de bases de datos distribuidas. APPC.LU6.2 es uno de los mecanismos de comunicación más importante para el éxito de la implementación de la arquitectura cliente/servidor.

En una red de trabajo IBM PC, la interface APPC/PC soporta el protocolo de nivel de sesión con un nivel de sincronización de confirmación. Este nivel permite conversaciones entre parejas que intercambian señales de confirmaciones/errores en requerimiento, asegurando así la confiabilidad de la transmisión de los datos. APPC/PC utiliza facilidades SNA, y está bien preparado para interconexiones entre una red IBM PC y una red SNA, controlado por un sistema de computador central (host).

NOVELL-NETWARE

NETWARE de NOVELL es uno de las más penetrantes implementaciones LAN, con un soporte directo o un protocolo comprensivo de pasarelas (gateway) sobre la mayoría de plataformas. NetWare puede ser usado con redes Ethernet, CSMA/CD (StarLan AT&T e IBM PC Network), así como arquitecturas token ring, que incluye token ring de IBM. NetWare emula NETBIOS, soporta archivos e impresoras compartidos, correo electrónico, acceso remoto, comunicación inter-LAN vía puente de NetWare, y pasarela (gateway) para SNA de IBM (NetWare SNA Gateway) sobre líneas de control de datos asíncronico (SDLC).

Netware proporciona un número de utilitarios y programas de monitoreo, que permite a los administradores de la red facilidades, para agregar usuarios, abrir/cerrar archivos, y darle mantenimiento al sistema de seguridad.

Adicionalmente a los vendedores de PC DOS, vendedores de UNIX ofrecen soporte directo de NetWare de Novell. Un número de vendedores de Hardware para UNIX han liberado o pronto liberarán, versiones de NetWare portable, y proporcionan más enlaces entre NetWare y host UNIX, NetWare portable es implementado como "no nativo" para estas plataformas de hardware host, pero mejor, éste corre como un huésped bajo otros sistemas operativos, como resultado una nueva y alta conectividad de gran nivel entre UNIX-NetWare, puede llegar a ser alcanzado, con el soporte de los clientes tradicionales de Netware (DOS, Windows, OS/2, Macintosh) disponible para host UNIX.

BANYAN-VINES

Sistemas Banyan (Westborough, MA) ofrecen uno de los sistemas operativos de red distribuido (NOS) más avanzados técnicamente en el mercado. Vines está diseñado para soportar grandes redes de PCs. Uno de los clientes más grandes del sistema Banyan es Compaq, corriendo sobre su red interna de trabajo, aproximadamente 11,000 PC dispersas geográficamente, usando VINES.

La arquitectura distribuida Vines integra directorios, seguridad y servicios de administración de la red, sobre servidores interconectados, cada uno de estos soportando una o más PC LANs (ver figura 5.6). Un servidor Vines puede correr sobre procesadores en PC basadas en INTEL 286,386,486, sobre una versión POSIX de sistema UNIX V. Vines soporta estaciones de trabajo cliente DOS, windows y OS/2.

La arquitectura VINES soporta múltiples tipos de LAN, incluyendo Ethernet y Token Ring, y virtualmente cualquier topología LAN. Vines proporciona soporte para una variedad de protocolos de comunicación, e incluye emuladores 3270, TCP/IP, y el protocolo de conmutación de paquetes X.25. Adicionalmente Vines puede correr sobre interconexiones de servidor a servidor en una red WAN, así proporcionando una vista global sencilla de la red de trabajo de una organización.

Vines proporciona un usuario con un panorama individual integrado a la red de trabajo. Vines usa mecanismos de autorización para soportar seguridad en una red amplia. Los administradores de la red son provistos con facilidad de configuración de la administración y con herramientas de monitoreo. Vines ofrece un conjunto de servicios estándar que incluyen servicios de archivos e impresión, así como un correo de red.

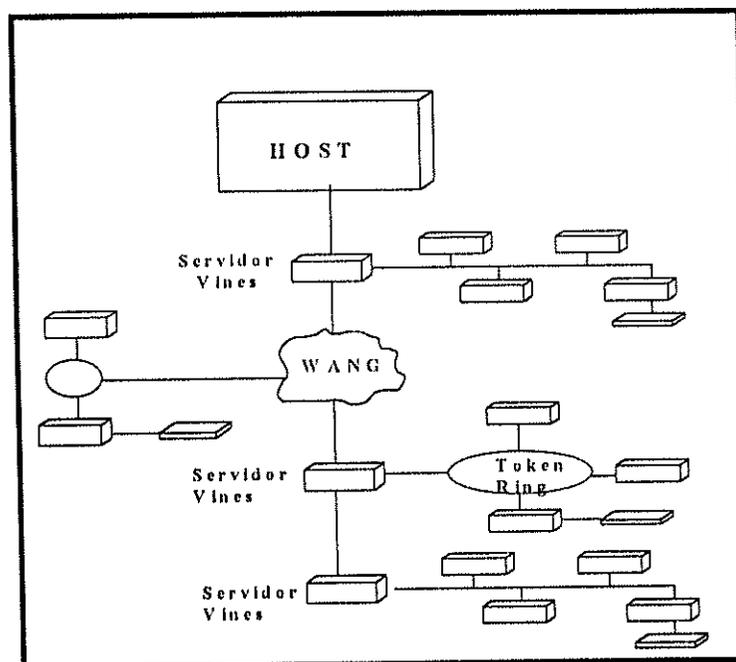


Fig. No. 5.6 Red Vines.

STARLAN DE AT&T

STARLAN de AT&T es una red de área local, que está basada en el protocolo estándar de IEEE 802.3 CSMA/CD, aunque ésta opera típicamente a tasas de transmisión menor que todas las del tipo Ethernet (1 MBits/Seg versus 10 MBits/Seg.). La red STARLAN está diseñada para operar sobre el cable telefónico estándar (alambre trenzado). STARLAN soporta ambos sistemas operativos DOS y UNIX en las siguientes combinaciones :

- Servidor DOS/Estación DOS
- Servidor UNIX/Estación UNIX
- Servidor UNIX/Estación DOS

La última es especialmente usada, cuando una organización construye una LAN conectando las PCs DOS existentes, a poderosos servidores UNIX.

STARLAN soporta una configuración de cadena de margarita (hasta 10 estaciones pueden ser conectadas a un cable individual), y la configuración estrella, la cual es construida al conectar segmentos de cadena de margarita (hasta 11 segmentos), a través de unidades de extensión de red de trabajo (hasta 12 unidades) en una estrella.

STARLAN proporciona servicios de compartimiento de archivos e impresoras, en ambos ambientes DOS y UNIX. Adicionalmente STARLAN puede emular NETBIOS, y proporcionar compatibilidad con los programas PC LAN de IBM.

3Com

La corporación 3Com es una de las mejores vendedores de LAN que ofrecen productos para IBM, Apple y líneas compatibles de computadoras personales. En los niveles físicos y control de enlace de datos, 3Com soporta el uso de formatos Ethernet y IEEE 802.3 CSMA/CD y tasa de datos hasta 10 Mbits/Seg. 3Com también soporta el protocolo token ring. Los productos 3Com LAN incluyen :

- Sistema operativo de red 3+Share, proporcionando compartimiento de archivos e impresoras.
- 3+Mail, Proporciona servicios de mensajería electrónica.
- Productos 3+Route, 3+Remote, y 3+NetConnect que permiten interconexión entre las redes 3Com.
- 3+3270, permitiendo a las redes 3Com acceder a un host SNA como una terminal 3270 de IBM.

OTRAS ARQUITECTURAS LAN

Adicionalmente a las arquitecturas LAN descritas, hay otras arquitecturas que son desarrolladas para aplicaciones específicas. Estas arquitecturas están basadas en un variedad de opciones permitidas por estándares, conforme los mismos son definidos por los entidades estándares como: IEEE, OSI, CCITT. Entre esas arquitecturas específicas de aplicación, son manufacturadas para arquitecturas LAN (MAP y TOP) y arquitecturas basadas en el uso de cable de fibra óptica (FDDI).

ESPECIFICACIONES MAP Y TOP

El protocolo de Automatización de Manufactura (MAP), se dirige al área de automatización de fabricas. Ha estado siendo desarrollado para tareas de fuerza, patrocinado por la General Motors. Está dirigido para usos de ingeniería, manufactura y automatización de la oficina en general. Boeing ha patrocinado otra tarea de fuerza, cuyo contrato fue desarrollar el Protocolo de oficina y Técnico (TOP).

En general, ambas MAP y TOP usan una estructura de red interconectada jerárquicamente. En la arquitectura jerárquica MAP/TOP, los niveles más bajos son ocupados por redes de área local. Dependiendo de la distancia involucrada, estas LANs pueden ser interconectadas sobre una red en un campus, en una red de área metropolitana (MAN) y una red de área amplia (WAN) .

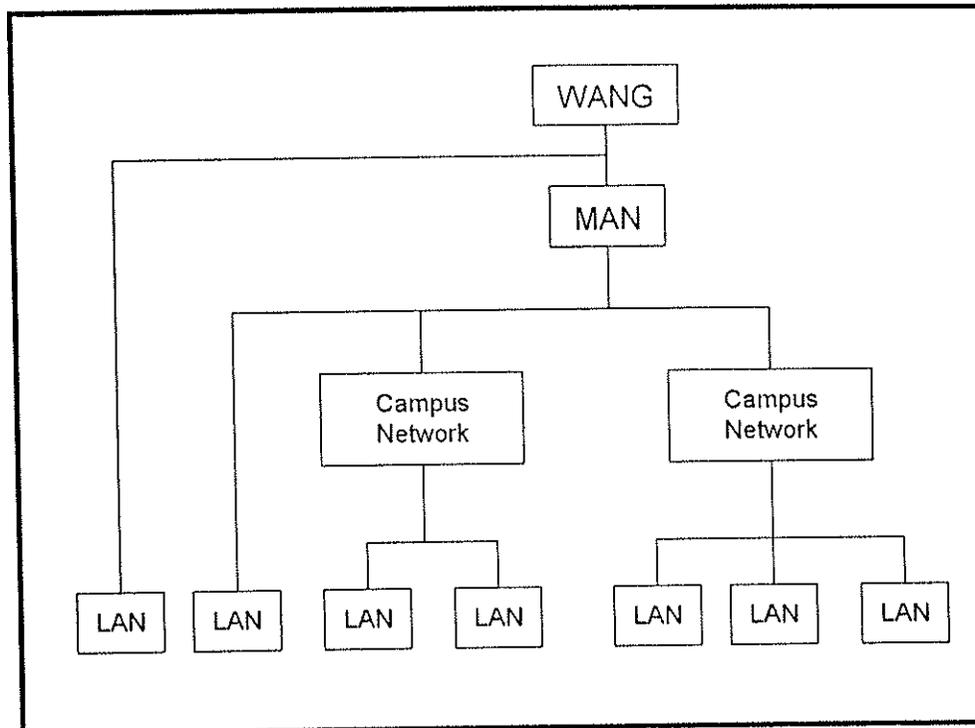


Fig. No. 5.7 Arquitectura Jerárquica MAP/TOP.

Tanto MAP como TOP usan las siete capas del modelo de referencia OSI como la base para sus respectivas arquitecturas. Las especificaciones MAP y TOP incluyen: un conjunto de los protocolos OSI, bien documentados, similares, los cuales permiten fácil interconexión entre redes MAP y TOP.

Ambos MAP y TOP usan el protocolo de control de enlace lógico IEEE 802.2, pero diferentes métodos de control de acceso. MAP usa el estándar token bus IEEE 802.4, mientras TOP ha seleccionado el estándar IEEE 802.3 CSMA/CD. En las capas superiores del modelo OSI, ambos el MAP y TOP soportan transmisión full-duplex y detección y recuperación de errores; ellos emplean la transferencia de archivos, el protocolo de administración y acceso, y una variedad de protocolos de servicios de aplicación.

FDDI

El estándar de interface de datos distribuidos en fibra (FDDI) fue desarrollado por, the American National Institute of Standards (ANSI). El comité X3T9.5 desarrollo el FDDI para conocer los requerimientos de los usuarios, de alto rendimiento de redes y gran velocidad en interconexión de redes.

El estándar FDDI ANSI está basado en el uso de la técnica de pase de token sobre cables de fibra óptica a tasas de 100 Mbits/Seg. El protocolo de pase de tokens usado en FDDI es similar al estándar token ring IEEE 802.5, aunque hay algunas pequeñas diferencias en el manejo del token. El estándar FDDI ha sido desarrollado para satisfacer los requerimientos de los siguientes tipos de red :

- Redes locales Backend son frecuentemente usadas para interconectar mainframes y grandes dispositivos de almacenamiento de datos. Tales interconexiones requieren tasas de transferencia de datos extremadamente altas.
- Redes locales Backbone usualmente son usadas para interconectar redes de trabajo de área local. Las redes de trabajo backbone necesitan tasas de transmisión altas, para eliminar un potencial cuello de botella en comunicaciones inter-LAN.
- Red de trabajo de oficina de alta velocidad son requeridas para soportar los requerimientos de altas tasas de transferencia de datos, para procesamiento de imágenes y gráficas. Las gráficas e imágenes de alta resolución son generalmente guardadas en un sistema llamado grandes objetos binarios (BLOBs), los cuales pueden ser de varios gigabytes de tamaño.

El estándar FDDI proporciona una especificación de red de trabajo confiable y de alta capacidad, reforzada por la conmutación de traslado óptico automático y configuración dual.

5.8 CONECTIVIDAD INTER-LAN

Las redes de área local permiten a un grupo de usuarios de computadoras comunicarse entre ellos, compartir información, servicios y dispositivos. Las LANs son ideales para ambientes de trabajo en grupo e implementaciones cliente/servidor, sobre una área geográfica pequeña. Sin embargo, en la realidad, usuarios de diferentes redes de área local necesitan comunicarse y compartir información, con miembros de otros grupos de trabajo. Por lo tanto, la interconectividad es frecuentemente requerida para interconectar las LANs existentes.

Las redes de área local pueden ser interconectadas directamente o a través de redes MANs y WANs de un elevado nivel. Cuando se interconectan las redes de área local o un segmento del LAN, varios métodos pueden ser usados :

- Los repetidores son un método sencillo para la interconexión. La funcionalidad de los repetidores está limitada a la interconexión de segmentos individuales de redes pequeñas, similares para formar una red grande. Típicamente los repetidores son usados en la topología LAN.
- Los puentes pueden interconectar redes que soportan diferentes protocolos de transmisión (LAN CSMA/CD y LAN token ring). Un puente, entre dos LAN, recibe mensajes de ambas redes, revisa los destinos, y transmite los mensaje a la LAN requerida. Desde que los mensajes son almacenados en el sistema del puente antes de retransmitirlo, los puentes deben estar informados de implementar la técnica de almacenamiento y reenvío.

Para manejar los mensajes propiamente, un puente debe de conocer y comprender el esquema de direccionamiento usado en ambas redes. Desde el punto de vista del modelo OSI, los puentes operan en la capa física y la capa de enlace.

- Los ruteadores desempeñan funciones de capa de red del modelo referencia OSI. Un ruteador rutea mensajes de un nodo a un nodo destino, a través de nodos intermedios. En este caso, dos direcciones deberfan acompañar el mensaje, las cuales son: el siguiente nodo en la ruta y nodo destino final. La

primera dirección cambia de una estación a otra a través de la ruta, mientras que la segunda dirección permanece constante. Para que un ruteador sea capaz de trabajar, las redes interconectadas deberfan de ser compatibles, la red de trabajo y las capas de alto nivel OSI (de la 3ra capa hacia arriba), pero pueden ser diferentes la capa física y la capa de enlace de datos. Una función clave de los ruteadores es determinar el siguiente nodo (estación) al cual será enviado el mensaje.

- Las pasarelas (gateways) son usadas para interconectar redes, que pueden ser construidas en arquitecturas completamente diferentes. Así las pasarelas (gateways) implementan funciones de las siete capas del modelo OSI. Los Gateways manejan cualquier mensaje, direcciones, y conversiones de protocolos necesarios para enviar un mensaje de una red a otra. Las pasarelas (gateways) ofrecen grandes flexibilidades en la interconexión de comunicaciones. El costo de esta flexibilidad es el pago de un precio más alto y un diseño, implementación, mantenimiento y operación de una pasarela (gateway) más compleja.

Ruteadores, puentes y pasarelas forman la base para la redes inter-LAN. La red backbone, vista como un candidato para la implementacion FDDI, es un buen ejemplo de solución de interconectividad de inter-LAN.

CAPITULO 6

APLICACION : DISEÑO DE UNA RED CLIENTE SERVIDOR

Para ilustrar los problemas potenciales de la distribución en un ambiente cliente/servidor, se hará el diseño de una red utilizando la arquitectura cliente/servidor para distribuir el procesamiento de la presentación, la lógica del negocio, de la administración de la información en la red, y aumentar el rendimiento de la red, para lograr tal propósito se utilizará la arquitectura multinivel. Se considera el siguiente escenario un banco comercial que actualmente tiene toda sus aplicaciones centralizadas en un host, con el objetivo de elevar el rendimiento del sistema en general se utiliza la arquitectura de tres niveles para lograr este objetivo.

Situación actual

En las oficinas centrales, se mantiene toda la información de las distintas aplicaciones en un host, para este diseño sólo se tomaron en cuenta las siguientes aplicaciones :

- Clientes
- Monetarios
- Ahorros
- Caja
- Contabilidad

Clientes : en esta aplicación se tiene la base de datos de los clientes, la programación de la aplicación; desde esta aplicación, se integra el cliente con los diversos productos y servicios que ofrece el banco, por ejemplo : cuentas que posee, análisis de reciprocidad del cliente, promedio de sus cuentas, calificación de cliente etc.

Monetarios : son la cuentas de cheques; esta aplicación administra la información de la cuentas, saldos, movimientos diarios, mensuales e históricos, control de chequeras, autorizaciones, sobregiros, los parámetros de toda la aplicación, por ejemplo : tasas, tipos de cuenta, tipos de transacciones, y todo el movimiento de la cuentas, ej. : pago de cheques, depósitos, notas de crédito/débito, autorizaciones varias, pago de intereses, etc.

Ahorros : se mantiene la información de la cuentas de ahorro, datos de la cuentas, saldos, movimientos diarios, mensuales e historia, control de libretas, parámetros de la aplicación por ejemplo: tasas, tipos de cuenta, tipos de transacciones, pago de intereses, etc.

Caja : esta aplicación se encuentra en el host central, e integra a todas las aplicaciones del banco; esta aplicación interactúa de manera transaccional con las otras aplicaciones (monetarios, ahorros, etc.), y afectan los saldos de las cuentas, por ejemplo: depósitos, retiros, pago de cheques etc. Además, lleva el control del efectivo, reportes de caja, y control de los cajeros, autorizaciones, etc.

Contabilidad : en esta aplicación se administra toda la contabilidad del banco, catálogo de cuentas, libros contables (diario, mayor, balance, estado de resultados); todas las aplicaciones interactúan con la contabilidad generando pólizas automáticas.

La arquitectura actual está basada en un host, el cual soporta todo el procesamiento de la lógica del negocio, administración de la información, presentación. En esta arquitectura, todos los componentes de la aplicación residen y son ejecutados sobre la plataforma de computación centralizada. Las agencias con las que cuenta el banco, están conectadas al computador central, por medio de líneas telefónicas dedicadas, para lo cual, se requiere de modems para lograr conectarse al host; en cada agencia, se dispone de varias terminales; para lograr que todas terminales utilicen el mismo medio de comunicación, se utiliza un multiplexor, el cual combina todas las señales de las terminales en una simple señal a ser transmitida y esta señal es conectada al módem, del lado de la central se utiliza otro modem que recibe la señal y ésta es conectada a un multiplexor para que la señal simple sea separada en las señales originales (ver figura 6.1).

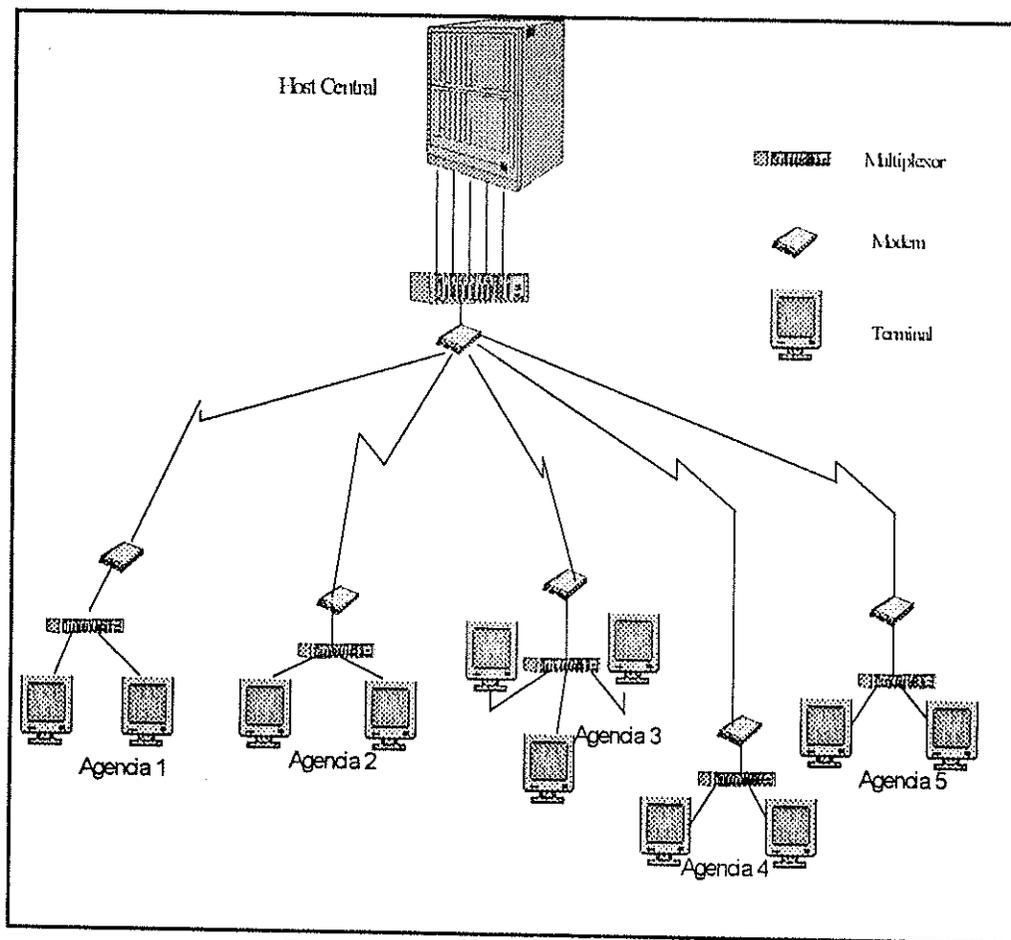


Figura No. 6.1 Red basada en un Host

DISEÑO DE LA RED UTILIZANDO LA ARQUITECTURA CLIENTE/SERVIDOR:

Al hacer el diseño con la arquitectura cliente/servidor, en cada agencia se introduce un servidor local y varios clientes, eliminando las terminales tontas. El servidor local en cada agencia está basado en poderosas computadoras pentium y Windows NT de Microsoft; los clientes están basados en poderosas computadoras personales y windows 95 de Microsoft. Para llevar a cabo esta implementación, todas las aplicaciones del banco se redefinieron de forma transaccional (para las aplicaciones que no estaban definidas de esta manera), por ejemplo, aperturas de clientes, apertura de cuentas, consultas de saldos, estados de cuenta etc.; éstas se realizan de manera transaccional como se muestra más adelante, es decir, que el requerimiento cliente y la respuesta a ese requerimiento está basado en transacciones.

Con el modelo cliente/servidor, se distribuyen los componentes de la aplicación:

- La lógica del procesamiento de presentación es trasladada hacia el cliente, y ésta ejecuta las siguientes tareas: formato de la pantalla, la lectura y escritura de la información de la pantalla, administración de las ventanas, el manejo del mouse, etc. La lógica de la presentación está basada totalmente en ambiente gráfico.
- La lógica del procesamiento del negocio está distribuida entre el host central y los clientes, ya que el código de la aplicación se encuentra en ambos lados.
- La lógica del procesamiento de la base de datos se encuentra distribuida en los clientes y el host central, y es código de la aplicación que manipula la información.

Para lograr la implementación de la nueva red, algunas aplicaciones fueron nuevamente diseñadas y programadas; el módulo de caja se trasladó del host central a cada servidor de agencia, y de esta manera se descarga al computador central. La nueva aplicación de caja se encuentra completamente integrada con las aplicaciones anteriores, y ésta interactúa de manera transaccional con cada una de estas, por ejemplo, una transacción de consulta de saldos de monetarios, la cual se define de la siguiente manera

Transacción 435 : consulta de saldos de cuenta de monetarios, la cual hace un requerimiento del saldo de la cuenta de manera transaccional al host central; para pedir el requerimiento, envía la siguiente información codificada :

- Número de transacción
- Tipo de servicio (Requerimiento)
- Número de cuenta

Cuando la transacción llega al host, éste procesa el tipo de transacción y establece que es un requerimiento de consulta de saldo de la cuenta de monetarios, obtiene la información deseada, la procesa y devuelve el requerimiento al cliente de la siguiente manera:

Devuelve la respuesta de la transacción 435

- Número de transacción
- Tipo de servicio (Respuesta)
- Número de cuenta

- Saldo de la cuenta
- Nombre de la cuenta

Todas las aplicaciones fueron rediseñadas, e implementadas tanto del lado del host y del cliente de manera similar.

Con este diseño, queda distribuido el procesamiento de la presentación, de la lógica y base de datos, entre los clientes y el host central; al efectuar este cambio el host central, distribuye el procesamiento, ya que de la forma anterior, todos los clientes son terminales tontas, es decir, que el computador central carga con todo el procesamiento de la aplicación. Al mismo tiempo, reduce la carga de la red de la siguiente manera: con el modelo anterior el procesamiento de la presentación se hace en el computador central, se arma toda la pantalla, ubicación de campos, etc., y se envía toda la pantalla armada en ambiente caracter a través de la red hacia la terminal tonta, al hacerlo de manera transaccional ya no envía toda la pantalla, si no solamente la información necesaria (transacción); con esto disminuye la carga de la red, y se aumenta su rendimiento; el nuevo diseño, el cliente recibe la respuesta de manera transaccional, la procesa y la presenta al usuario en un ambiente gráfico, al mismo tiempo que todo el procesamiento de la presentación es local en cada cliente. Debido a que los clientes se encuentran en un ambiente Windows, éstos consumen una gran cantidad de recursos (memoria, disco, procesamiento) que el computador central no podría soportar, además que este sólo se maneja en ambiente caracter.

El diseño de la nueva red se observa en la figura 6.2, y tiene los siguientes componentes:

- Servidores locales en cada agencia
- Clientes inteligentes (computadoras personales)
- Hub recibe señales de una estación y con precisión la retransmite a otra estación
- Los ruteadores interconecta red con otras redes, proporciona el control del tráfico y funciones de filtración cuando más de un camino existe entre dos puntos finales en la red, direcciona los paquetes en el camino más eficiente o económico en la red.
- Modems, es el que convierte las señales digitales de las computadoras para ser transmitida sobre una línea de teléfono análoga.
- Se utiliza el protocolo de comunicación TCP/IP, por las aplicaciones que este puede manejar, para transferir archivos, para correo electrónico, para el manejo de archivos de la red (NFS), para administrar la red, ya que con este protocolo desde cualquier lugar puede acceder servicios e información a cualquier otro nodo en la red. También porque es un protocolo estándar y ampliamente usado, y por una futura integración con Internet.
- FDDI (interface de datos distribuidos en fibra) se utiliza para construir el backbone¹⁵ e la red, transmitiendo a altas velocidades.

¹⁵ backbone : es una red que conecta dos o más segmentos redes de área local o sub redes y proporciona el camino para transmitir paquetes de información entre ellos.

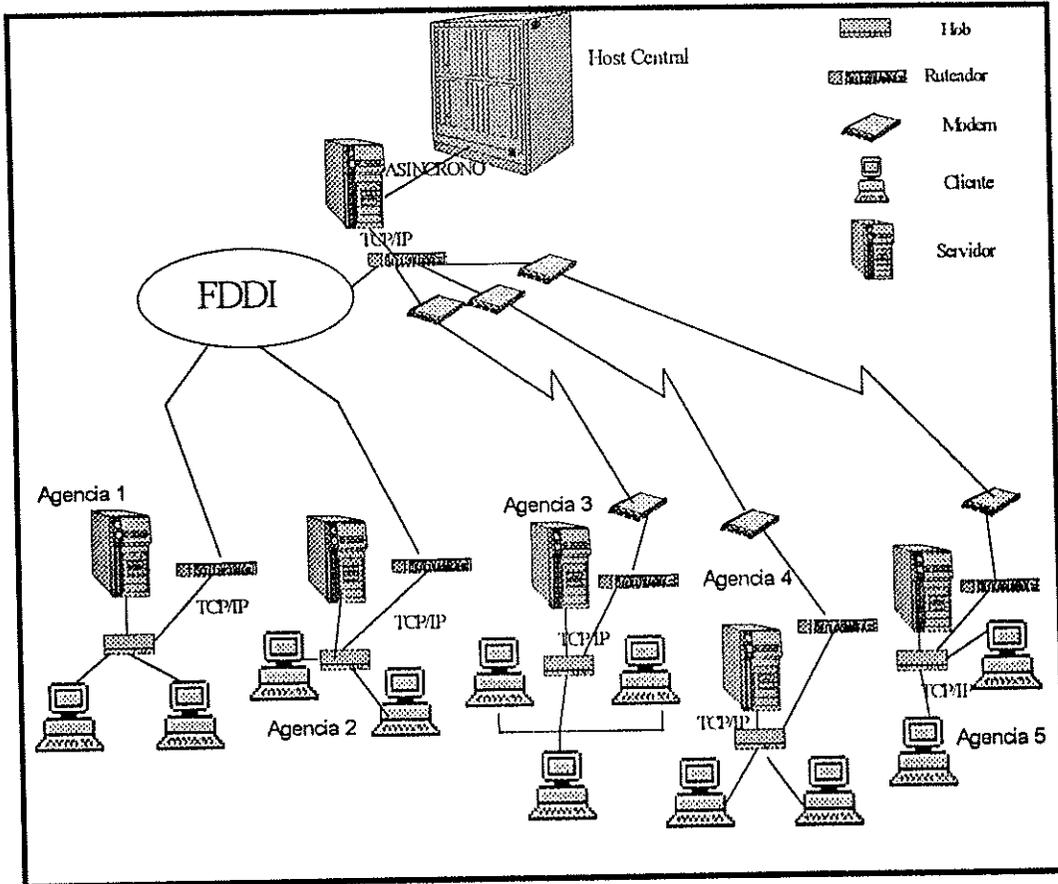


Fig. No. 6.2 Red Cliente/Servidor

CONCLUSIONES

1. El modelo de procesamiento cliente/servidor es una extensión natural del procesamiento de dispositivos compartidos. La principal razón para el cambio fue que, en un gran ambiente LAN, compartir servicios de archivos e impresión, entre las estaciones de trabajo en un grupo de LAN, representaba solamente una fracción de una aplicación típica. La parte significativa de la funcionalidad de la aplicación fue también un buen candidato para compartir entre usuarios de LAN; así algunas de las aplicaciones de procesamiento, fueron distribuidas a nuevos servidores; el servidor que recibe requerimientos desde aplicaciones que están corriendo en estaciones de trabajo (clientes) y tiene procesos para cada uno de los clientes.
2. El modelo cliente/servidor permite que el procesamiento resida cerca de la fuente de los datos, que están siendo procesados, por lo tanto, el tráfico de la red y el tiempo de respuesta puede ser reducido en gran manera, e incrementar la capacidad de transportar sobre la red una carga pesada. Por lo tanto, los requerimientos del ancho de la banda de las redes y su consecuente costo pueden ser reducidos.
3. Es muy probable que la implementación del modelo cliente/servidor reducirá el costo de mantenimiento del software, incrementará la portabilidad del software y el rendimiento de las redes existentes, y aun más eliminará las aplicaciones acceso remoto (backlog), por el incremento de la productividad de los desarrolladores y reducen el ciclo de vida del desarrollo.
4. Los ambientes de computación distribuido representan una contribución invaluable a los desarrolladores de computación distribuida abierta, para hacer de los ambientes de computación distribuida una arquitectura completa.
5. Los ambientes de computación abiertos no significan la eliminación de sistemas operativos propietarios. Realmente, los sistemas abiertos pueden coexistir con los sistemas operativos propietarios. Esta coexistencia debe ir acompañada con puntos de interfaces y conexión, Realmente, los ambientes de computación abiertos benefician a los proveedores de sistemas, a los proveedores de software independiente y a los usuarios finales.
6. En el procesamiento basado en un Host, los componentes de esta aplicación residen en el mismo sistema y son combinados en un programa ejecutable. La distribución no está teniendo lugar y, en general, es restringida por los recursos limitados de la plataforma en la cual es ejecutada. Con la llegada de la computación distribuida, nuevas oportunidades han sido abiertas para los desarrolladores de los sistemas y los usuarios finales; aplicaciones escalables y portables son capaces de correr sobre redes de sistemas abiertos que son transparentes a los usuarios finales, y que ahora pueden ser desarrolladas. Debido a la distribución de los recursos a través de la red, el costo-beneficio puede llegar a ser significativo.
7. No todos los negocios necesitan un poder y la capacidad de almacenamiento de computación de un mainframe, ni todos los negocios y las aplicaciones más importantes justifican la millonaria inversión de un mainframe y el costo elevado para mantener la información centralizada.

RECOMENDACIONES

1. Al distribuir la lógica de la aplicación entre los clientes y servidores, se deben de realizar con mucho cuidado, ya que una distribución errónea de la lógica de la aplicación, puede llegar a provocar un cuello de botella en la misma forma que un computador central en la arquitectura maestro-esclavo.
2. Las aplicaciones distribuidas, especialmente aquellas que son diseñadas para procesamiento cooperativo, son más complejas que las no distribuidas; esto es cierto para el desarrollo de aplicaciones en ambientes de corrida real, y en las herramientas usadas para el manejo de estos ambientes distribuidos. Sin embargo, algo de esta complejidad puede ser eliminada reduciendo un problema grande al convertirlo en un conjunto de problemas más pequeños, posiblemente problemas interdependientes, similares al diseño del sistema modular.
3. Si el volumen de la transacción y todos los requerimientos son conocidos, el costo de las transacciones por segundo puede ayudar a determinar cuánto hay que gastar para soportar una aplicación de negocios en particular.
4. Al diseñar redes de trabajo de área local, muchos factores interdependientes deben de ser tomados en consideración; los medios de transmisión, métodos de control de transmisión y acceso, topología de red de trabajo, ancho de banda, y tasas de información. Todos estos factores afectan el desempeño y costo de la red de trabajo. Las decisiones concernientes a la topología de red de trabajo, control de transmisión y métodos de control de acceso deberían de tomarse basados en los requerimientos de procesamiento y costo de un LAN particular.