

11
2ej



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS "ARAGON"

FALLA DE ORIGEN

LA ARQUITECTURA CLIENTE/SERVIDOR HACIA LA
TENDENCIA DOWNSIZE

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A N :
LETICIA CRUZ JUAREZ
JOSE ALBERTO VALERIANO VILLEGAS

ASESOR: ING. DONACIANO JIMENEZ VAZQUEZ



MEXICO, D. F.

SEPTIEMBRE 1995



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

LA ARQUITECTURA

CLIENTE/SERVIDOR

HACIA LA TENDENCIA

DOWNSIZE

LA ARQUITECTURA CLIENTE/SERVIDOR HACIA LA TENDENCIA DOWNSIZE

CONTENIDO

- CARTAS DE ACEPTACION
- AGRADECIMIENTOS
- PROLOGO
- OBJETIVO
- INTRODUCCION

CAPITULO I: INTRODUCCION A LAS REDES

1.1 BREVE HISTORIA DE LAS REDES LOCALES.	1
1.2 COMPUTO CLIENTE/SERVIDOR.	2
1.3 VENTAJAS DEL COMPUTO CLIENTE/SERVIDOR.	3
1.4 DESVENTAJAS DEL COMPUTO CLIENTE/SERVIDOR	7
1.5 EL MODELO OSI	7
1.5.1. LOS 7 NIVELES DEL MODELO OSI	8
1.5.2. FUNCIONAMIENTO DEL MODELO OSI	11
1.6. DOWNSIZE.	13
1.6.1. INCREMENTO DE PRODUCTIVIDAD	13
1.6.2. EL MODELO CLIENTE/SERVIDOR EN LAS ORGANIZACIONES	14

CAPITULO II: ELEMENTOS Y ESTRUCTURAS DE REDES LAN

2.1 MEDIOS DE COMUNICACION	15
2.1.1 CABLE COAXIAL	15
2.1.2. CABLE TELEFONICO	16
2.1.3. FIBRA OPTICA:	17
2.2 TOPOLOGIA DE REDES	18
2.2.1. BUS LINEAL	18
2.2.2. TOPOLOGIA DE ESTRELLA	19
2.2.3. TOPOLOGIA DE ANILLO	19

2.3 TARJETAS DE INTERFACES DE RED	20
2.4 ESTANDAR IEEE 802, FDDI y X.25	21
2.4.1. ETHERNET: RED LOCAL CSMA/CD	23
2.4.2. TOKEN RING (PASO DE TESTIGO EN ANILLO)	25
2.4.3. INTERFAZ DE DATOS DISTRIBUIDA DE FIBRA (FDDI)	27
2.4.4. PROTOCOLO X.25	28
2.5 PROTOCOLOS DE NIVELES SUPERIORES	31
2.7.1. TCP/IP	32
2.7.2. NetBIOS	36
2.6 PUENTES Y RUTEADORES	38
2.6.1. PUENTES	38
2.6.2. RUTEADORES	39

CAPITULO 3 LA ARQUITECTURA CLIENTE/SERVIDOR

Sistemas de Procesos Distribuidos

Introducción	43
3.1 PROCESOS DISTRIBUIDOS	43
3.1.1. DEFINICION DE PROCESOS DISTRIBUIDOS	43
3.1.2. MODELOS DE PROCESOS DISTRIBUIDOS	43
3.1.3. ESTRUCTURA DE SISTEMAS DISTRIBUIDOS	44
3.1.4. DISTRIBUCION DE LA ARQUITECTURA CLIENTE/SERVIDOR	46
3.2 CLIENTE/SERVIDOR	49
3.2.1. ATRIBUTOS Y COMPONENTES DEL CLIENTE	50
3.2.2. ATRIBUTOS Y COMPONENTES DEL SERVIDOR	51
3.3 COMUNICACIONES	55
3.3.1. ASPECTOS DE CONEXIÓN	57
3.3.2. ASPECTOS DE DIRECCIONAMIENTO	58
3.3.3. ASPECTOS DE SINCRONIZACION.	62
3.4 LLAMADAS A PROCEDIMIENTOS REMOTOS	62
3.4.1. PRINCIPALES COMPONENTES DE UN RPC	64
3.4.2. MODELOS DE INTERACCION	66
3.4.3. TIPOS DE FALLAS	67
3.4.4. SEGURIDAD	68

CAPITULO IV: SISTEMAS OPERATIVOS DE RED

INTRODUCCION	69
4.1 PARTES PRINCIPALES DE UN SISTEMA OPERATIVO:	70
4.2 MODELOS DE SISTEMAS OPERATIVOS	71
4.2.1. MODELO MONOLITICO	71
4.1.2. SISTEMAS CON CAPAS	73
4.1.3. MODELO CLIENTE/SERVIDOR	73
4.3 PROCESOS DISTRIBUIDOS	74
4.3.1. COMUNICACION ENTRE PROCESOS (IPC)	75
4.3.2. HILOS (THREADS)	77
4.4 ADMINISTRACION DE LA MEMORIA	80
4.4.1. MANEJO DE MEMORIA POR PARTICIONES (FRAGMENTACION)	80
4.4.2. PARTICIONES RELOCALIZABLES	80
4.4.3. SWAPPING (INTERCAMBIO)	81
4.4.4. PAGINACION	81
4.4.5. PAGINACIÓN POR DEMANDA.	82
4.4.6. SEGMENTACION.	82
4.4.7. SEGMENTACION-PAGINACION	82
4.5 ARCHIVOS	83
4.5.1. SISTEMAS DE ARCHIVOS DISTRIBUIDOS	83
4.5.2. LA INTERFAZ DEL SERVIDOR DE DIRECTORIOS.	84
4.5.2. UN EJEMPLO: EL SISTEMA DE ARCHIVOS ANDREW.	85
4.6 SISTEMAS OPERATIVOS DE RED	88
4.6.1 WINDOWS NT	88

CAPITULO V: ADMINISTRACION DE REDES

INTRODUCCION	97
5.1 ADMINISTRACION DE SISTEMAS.	97
5.1.1. OBJETIVOS FUNCIONALES	97
5.1.2 METODO PARA LA ADMINISTRACION DE SISTEMAS	98
5.2 ESTRUCTURAS DE LA ADMINISTRACION DE SISTEMAS	99
5.2.1. ESTRUCTURA MULTIDOMINIO	100
5.2.2. ESTRUCTURA DEL NODO DE SISTEMA DE ADMINISTRACION	101

5.3 ADMINISTRACION DE APLICACIONES	102
5.3.1 ADMINISTRACION DE FALLAS	102
5.3.2. CONFIGURACION Y ADMINISTRACION DE OPERACIONES.	104
5.3.3 ADMINISTRACION DEL DESEMPEÑO	106
5.3.4 ADMINISTRACION DE COSTOS Y USO DE LA RED	107
5.3.5 ADMINISTRACION DE SEGURIDAD	108
5.3.6 ENCRIPAMIENTO DE DATOS	109
5.4 RESPALDOS	111
5.4.1. RESPALDOS SOBRE LA RED	111
5.4.2. RESPALDO CON SERVIDORES DE ARCHIVO	112
5.4.3. RESPALDOS SIN OPERADOR	113
5.4.4. ADMINISTRACION DE RESPALDOS	113
5.5 SOFTWARE PARA LA ADMINISTRACION DE REDES	113
5.5.1. PRODUCTOS PARA LA ADMINISTRACION	113
5.5.2. PROTOCOLO SIMPLE PARA LA ADMINISTRACIÓN DE REDES (SMNP)	114
5.5.3. FUNCIONAMIENTO DE SNMP	116
5.5.4. PROTOCOLO COMUN DE INFORMACION DE ADMINISTRACION (CIMP)	118

CAPITULO VI: APLICACIONES

6.1 BASE DE DATOS SQL	119
6.1.1 FUNCIONES DE SQL	116
6.1.2. SQL EN RED.	120
6.1.3 SERVIDOR DE BASE DE DATOS.	122
6.1.4. ARQUITECTURAS DE BASES DE DATOS	122
6.1.5 INTEROPERABILIDAD	124
6.2. MONITORES TP	128
6.3 GROUPWARE	130
6.4 OBJETOS DISTRIBUIDOS	131
6.4 DESARROLLO DE SECOFI	132

- **CONCLUSIONES**

- **BIBLIOGRAFIA**



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ARAGÓN
DIRECCION

LETICIA CRUZ JUAREZ
P R E S E N T E .

En contestación a la solicitud de fecha 26 de abril del año en curso, presentada por José Alberto Valeriano Villegas y usted relativa a la autorización que se les debe conceder para que el señor profesor, Ing. DONACIANO JIMENEZ VAZQUEZ pueda dirigirles el trabajo de Tesis denominado " LA ARQUITECTURA CLIENTE-SERVIDOR HACIA LA TENDENCIA DOWNSIZE ", con fundamento en el punto 6 y siguientes, del Reglamento para Exámenes Profesionales en esta Escuela, y toda vez que la documentación presentada por usted reúne los requisitos que establece el precitado Reglamento; me permito comunicarle que ha sido aprobada su solicitud.

Aprovecho la ocasión para reiterarle mi distinguida consideración.

ATENTAMENTE
"POR MI RAZA HABLARA EL ESPañOL"
San Juan de Aragón, Mex., 2 de mayo de 1994
EL DIRECTOR
M. EN C. CLAUDIO C. MERRIFLECA CASTRO



c c p Lic. Alberto Ibarra Rosas, Jefe de la Unidad Académica.
c c p Ing. Silvia Vega Muytoy, Jefe de la Carrera de Ingeniería
en Computación.
c c p Ing. Donaciano Jiménez Vázquez, Asesor de Tesis.

CCMC'AIR'11a.



UNIVERSIDAD NACIONAL
AVENIDA DE
MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ARAGÓN
DIRECCION

JOSE ALBERTO VALERIANO VILLEGAS
P R E S E N T E .

En contestación a la solicitud de fecha 26 de abril del año en curso, presentada por Leticia Cruz Juárez y usted, relativa a la autorización que se les debe conceder para que el señor profesor, Ing. DONACIANO JIMENEZ VAZQUEZ pueda dirigirles el trabajo de Tesis denominado " LA ARQUITECTURA CLIENTE-SERVIDOR HACIA LA TENDENCIA DOWNSIZE ", con fundamento en el punto 6 y siguientes, del Reglamento para Exámenes Profesionales en esta Escuela, y toda vez que la documentación presentada por usted reúne los requisitos que establece el precitado Reglamento; me permito comunicarle que ha sido aprobada su solicitud.

Aprovecho la ocasión para reiterarle mi distinguida consideración.

ATENTAMENTE
"POR MI RAZA HABLARA EL ESPÍRITU"
San Juan de Aragón, Mex., 2 de mayo de 1987
EL DIRECTOR

M en V. CRISTÓBAL C. MERRIFIELD CASTRO



[Handwritten signature]
[Handwritten signature]

- c c p Lic. Alberto Ibarra Rosas, Jefe de la Unidad Académica.
- c c p Ing. Silvia Vega Muytoy, Jefe de la Carrera de Ingeniería en Computación.
- c c p Ing. Donaciano Jiménez Vázquez, Asesor de Tesis.

CCMC'AIR'11a.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
CAMPUS - ARAGÓN
JEFATURA DE CARRERA DE INGENIERIA EN COMPUTACION

- ING. DONACIANO JIMENEZ VAZQUEZ
- ING. FERNANDO MARQUEZ CHAVEZ
- ING. LILIA ENCISO GARCIA
- ING. SILVIA VEGA MUYTOY
- ING. JUAN GASTALDI PEREZ

[Handwritten signatures and initials over the list of names]

Informamos a ustedes de la autorización que se le concede a la alumna LETICIA CRUZ JUAREZ, para desarrollar en forma conjunta con JOSE ALBERTO VALERIANO VILLEGAS, el trabajo de tesis "LA ARQUITECTURA CLIENTE-SERVIDOR HACIA LA TENDENCIA DOWNSIZE", dirigida por el Ing. Donaciano Jiménez Vázquez, solicitando a ustedes, sean tan amables de revisar el avance del mismo y hacer las observaciones que consideren pertinentes, o en su caso, indicar a la alumna si dicha revisión se hará a la conclusión del trabajo de tesis.

Sin otro particular, aprovecho la ocasión para enviarles un cordial saludo.

ATENTAMENTE

"POR MI RAZA HABLARA EL ESPERANTO"

San Juan de Aragón, Edo. de Méx., Junio 21 de 1985.

JEFE DE CARRERA

ING. SILVIA VEGA MUYTOY

[Handwritten signature of Silvia Vega Muytoy]



SVM/Tbh.

A MARY, PEPE Y MIS HEMANOS POR EL APOYO DE TODA LA VIDA.

A LA BANDA DE LOS "CHIPOTLES" POR TODOS LOS BUENO MOMENTOS

LETY JUNTOS LO LOGRAMOS

ALBERTO

AGRADESCO A *DIOS*, EL HABERME DADO LA OPORTUNIDAD DE TENER LA VIDA, Y TODO LO BUENO DE ESTE MUNDO.

GRACIAS MAMA POR HABERME DADO LA VIDA, POR TODA LA FELICIDAD Y APOYO QUE ME HAS BRINDASTE PARA LLEGAR HASTA DONDE ME ENCUENTRO AHORITA. GRACIAS MADRECITA POR TODO ESTO Y MAS...

GRACIAS PAPA POR TODO EL APOYO QUE ME BRINDO DESDE LO MAS PROFUNDO DE TU SER Y DE TODA LA FELICIDAD QUE ME DISTE DESDE SIEMPRE. GRACIAS PAPA, DONDE QUIERA QUE TE ENCUENTRES.

GRACIAS QUERIDOS HERMANOS, POR TODOS LOS MOMENTOS TAN AGRADABLES QUE ME HAN DADO Y POR TODO SU APOYO BRINDADO.

DEDICO ESTE TRABAJO CON MUCHO CARÍÑO A TODOS MIS SOBRINOS: TOMASITO, TOÑITO, MIGUELITO, CRISTINA, JESSICA Y LOS QUE VIENE PARA QUE ALGUN DIA LLEGUEN A REALIZAR UN TRABAJO SIMILAR O MEJOR. GRACIAS POR TODA SU TERNURA.

CUANDO ESTAS A PUNTO DE DARTE POR VENCIDO Y OYES LAS PALABRAS: "QUE PASO REYNA, NO TE PUEDES DOBLAR AHORA". GRACIAS JOSE, POR ESTAS PALABRAS TAN ALENTADORAS Y POR ENSEÑARME A ENCONTRARLE LA CHISPA A LA VIDA.

AGRADESCO A TODA LA GENTE QUE ESTUVO A MI ALREDEDOR, SIEMPRE APOYANDO DESINTERESADAMENTE. A TODOS MIS AMIGOS Y COMPAÑEROS, EN FORMA ESPECIAL A OLGA, CARMEN, CLAUDIA, ANDREA, JESUS Y RUBEN.

GRACIAS A TODOS LOS QUE HAN ESTADO CONMIGO Y A LOS QUE YA NO SE ENCUENTRA Y SIEMPRE ME ESTUVIERON DANDO VALOR PARA SEGUIR LUCHANDO. GRACIAS ABUELITA JUVE.

GRACIAS ALBERTO, POR LA AYUDA QUE ME BRINDASTE DESDE EL PRIMER SEMESTRE HASTA EL FINAL, Y POR QUE SIEMPRE FUISTE MUY COMPENSIVO CONMIGO.

GACIAS A TODOS.

LETY

AGRADESEMOS A LA UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO EL HABERNOS DADO LA OPORTUNIDAD DE SUPERARNOS EN LA GLORIOSA ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES PLANTEL "ARAGON".

AGRADECEMOS A TODOS NUESTROS MAESTROS QUE DESDE NUESTRA INFANCIA, NOS ENSEÑARON EL CAMINO DE LA SUPERACION, SOBRE TODO A LOS PROFESORES DE LA ENEP. Y DE MANERA MUY ESPECIAL A LOS PROFESORES:

- **ING. DONACIANO JIMENEZ VAZQUEZ**
- **ING. JUAN GASTALDI PEREZ**
- **ING. SILVIA VEGA MONTOY**

POR TODO SU APOYO BRINDADO PARA LLEGAR A LA CULMINACION DE UNA META MAS CUMPLIDA.

LETICIA Y JOSE ALBERTO

PROLOGO

Durante nuestra estancia como estudiantes en la Escuela Nacional de Estudios Profesionales "Aragón", de la Universidad Nacional Autónoma de México, notamos que una de las principales preocupaciones de los profesores de la carrera de Ingeniería en Computación, es poder dar a los alumnos, no solamente los conocimientos de las tareas fundamentales de esta interesante disciplina, si no también, que conociéramos los últimos avances tecnológicos en la mayoría de las áreas que involucran nuestra carrera profesional.

Desgraciadamente por múltiples razones, que sería estéril mencionar, esta preocupación de los profesores, es verdad que se esta cristalizando pero de manera lenta.

Al egresar de la carrera de Ingeniería en Computación e integramos al mercado de trabajo pudimos palpar la importancia de estar al tanto de conocimientos, fundamentos y avances en tecnología, utilizada indispensablemente para desarrollarnos profesionalmente en nuestra área de trabajo.

Son muy diversas las áreas de conocimiento que se involucran con la carrera de Ingeniería en Computación, y una de las que se presentan, no solo con un gran auge evolutivo, si no también, en pleno crecimiento en cuanto a uso, son las redes de computadoras. Estas se presentan de diversas arquitecturas, teniendo las mas actual: La arquitectura Cliente/Servidor.

De los anterior nace la inquietud de poder conocer mas de las redes de computadoras que utiliza el modelo Cliente/Servidor. Además otra inquietud de nosotros fue el poder contribuir en una pequeña parte a ese cambio en nuestra Escuela, para que las generaciones posteriores disponga de material de la Arquitectura Cliente/Servidor, tan mencionado hoy en día.

Es por esta razón que nos dimos a la tarea de realizar un estudio descriptivo de la arquitectura de redes de computadoras basándonos en la Arquitectura Cliente/Servidor, por ser el modelo mas utilizado actualmente, por seguir la tendencia de reducción de tamaño de equipo de computo, conocido como Downsize.

JOSE ALBERTO Y LETICIA

OBJETIVO:

REALIZAR UN ESTUDIO DE LAS REDES LAN, BASANDOSE EN LA ARQUITECTURA CLIENTE/SERVIDOR, DANDO UNA IDEA CLARA DE LOS ELEMENTOS QUE LA COMPONEN, LA FORMA EN QUE SE CONSTRUYEN Y SU FUNCIONAMIENTO BAJO ESTE MODELO.

REFLEJAR POR QUE ESTA TECNOLOGIA SE PRESENTA COMO UNA ALTERNATIVA ADECUADA A LAS NECESIDADES ACTUALES DE PROCESAMIENTO DE INFORMACION.

INTRODUCCION

Se dice frecuentemente que la década de los 90's es la "era de la información". Existen sin embargo, diversos puntos de vista sobre la manera en que debería efectuarse el procesamiento de datos, aunque se coincide en que nos encontramos en medio de un proceso de evolución que se prolongará todavía por algunos años y que cambiará la forma de obtener y utilizar la información almacenada electrónicamente.

Los sistemas de computación han evolucionado de tal modo que ya proporcionan una variedad de formas para procesamiento distribuido que simplifica la ejecución, desarrollo de aplicaciones y maximizar el uso de los recursos de la red corporativa.

Una arquitectura es un conjunto de definiciones, reglas y términos, que se emplean para construir un producto. En consecuencia, la arquitectura cliente/servidor se apega a estos principios. Cliente/servidor fundamentalmente es un tipo de cómputo distribuido y cómputo cooperativo. Los actores, clientes y servidores, establecen una comunicación entre pares (peer-to-peer), o sea, de igual a igual dado que los clientes de una aplicación pueden actuar como servidores de otra.

Los equipos multiusuarios empezaron como un anfitrión con terminales tontas conectadas a ellos. En esta configuración, las terminales tontas actúan sólo como estaciones de entrada/salida y sin ninguna capacidad de cómputo asociada, dejando al anfitrión toda la carga de proceso y control. Los equipos maestro/esclavo constituyeron una etapa posterior en que la computadora maestra coordinaba el trabajo de los equipos esclavos, los que a su vez se encargaban de los equipos periféricos. La ventaja principal de esta configuración es que descarga el trabajo de entrada/salida a los equipos esclavos, pudiéndose dedicar entonces el equipo maestro al procesamiento propiamente dicho. La tercer etapa en los ambientes de cómputo es la cliente/servidor, en ella el cómputo se divide entre el servidor y el cliente, aprovechando mejor la potencia de cómputo de los equipos existentes hoy en día.

La arquitectura computacional cliente/servidor, es la primera solución en tecnología informática que satisface las presiones de costo/rendimiento de una empresa actual de alto nivel. Es por esto que se observa una clara tendencia de migración hacia esta arquitectura.

El cliente/servidor es una etapa más en el proceso del *downsizing*. Las redes de computadoras modernas se basan en la idea de disponer la funciones y *protocolos* (se explican los protocolos en el capítulo 2) en varios niveles, para conseguir los siguientes objetivos:

-
- Descomponer lógicamente una red compleja en partes (capas o niveles) más pequeñas y fáciles de entender, proporcionando interfaces normalizados entre las distintas funciones de la red
 - Conseguir simetría en las funciones que se realizan en cada nodo de la red, cada nivel ha de llevar a cabo las mismas funciones que su contrapartida en otros nodos de la red.
 - Establecer un lenguaje normalizado que permita clarificar la comunicación entre los distintos diseñadores, fabricantes, distribuidores y usuarios de redes, a la hora de discutir las funciones de una red.

El modelo OSI de ISO (International Standards Organization) intenta plasmar todas las ideas anteriores, por lo cual se le dedica una sección en este capítulo para explicarlo a detalle. Además haremos referencia en varias ocasiones en capítulos posteriores.

Teniendo en mente los conceptos expresados en las líneas anteriores el presente trabajo seguirá estas guías como directrices para explicar la arquitectura cliente/servidor a través de sus partes dado que el modelo cliente/servidor no es un objeto, sino una arquitectura como su nombre lo indica.

La arquitectura cliente/servidor seguirá evolucionando y tomando presencia en el mercado.

-
- Descomponer lógicamente una red compleja en partes (capas o niveles) más pequeñas y fáciles de entender, proporcionando interfaces normalizados entre las distintas funciones de la red
 - Conseguir simetría en las funciones que se realizan en cada nodo de la red. cada nivel ha de llevar a cabo las mismas funciones que su contrapartida en otros nodos de la red.
 - Establecer un lenguaje normalizado que permita clarificar la comunicación entre los distintos diseñadores, fabricantes, distribuidores y usuarios de redes, a la hora de discutir las funciones de una red.

El modelo OSI de ISO (International Standards Organization) intenta plasmar todas las ideas anteriores, por lo cual se le dedica una sección en este capítulo para explicarlo a detalle. Además haremos referencia en varias ocasiones en capítulos posteriores.

Teniendo en mente los conceptos expresados en las líneas anteriores el presente trabajo seguirá estas guías como directrices para explicar la arquitectura cliente/servidor a través de sus partes dado que el modelo cliente/servidor no es un objeto, sino una arquitectura como su nombre lo indica.

La arquitectura cliente/servidor seguirá evolucionando y tomando presencia en el mercado.

CAPITULO I

INTRODUCCION

A LAS REDES

CAPITULO I: INTRODUCCION A LAS REDES

La arquitectura cliente/servidor es un modelo que consta de tres partes principalmente, que son cliente, servidor y equipo de comunicaciones, resultando una representación de un sistema individual. Aunque el modelo cliente/servidor se puede implementar de distintas formas, la forma más común de implementarla es utilizando redes de area local (LAN). Ya que las computadoras llegaron a ser una parte verdaderamente útil de la sociedad cuando se encuentran enlazadas por una infraestructura, tal como la red carretera o la red eléctrica, creando una nueva clase de mercado para servicios de información.

1.1 BREVE HISTORIA DE LAS REDES LOCALES.

El almacenamiento y el análisis de la información ha sido uno de los grandes problemas a los que se ha enfrentado el hombre desde que inventó la escritura. No es sino hasta la segunda mitad del siglo XX que ha podido resolver, parcialmente, este problema con la invención de la computadora.

En la década de los 50's el hombre dio un gran salto al inventar la computadora electrónica. La información ya podía enviarse en grandes cantidades a un lugar central donde se realizaba su procesamiento. ahora el problema era que esta información (que se encontraba en grandes cajas repletas de tarjetas) tenía que ser "acarreada" al departamento de proceso de datos.

Con la aparición de las terminales en la década de los 60's, se logró una comunicación directa, y por lo tanto más rápida y eficiente, entre los usuarios y la unidad central de proceso, pero se encontró un obstáculo: entre más terminales y otros periféricos se agregaban al computador central, decaía la velocidad de comunicación. En la figura 1.1 se muestra los distintos tipos de máquina que se usaban antes y las que se utilizan en la actualidad.

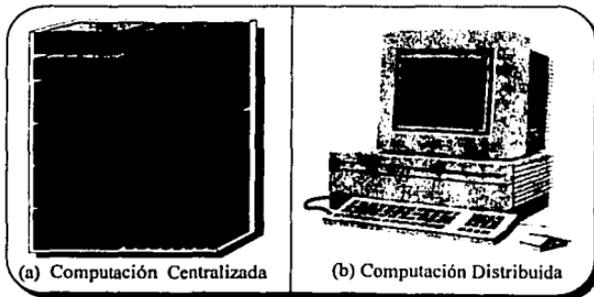


Figura 1.1 Formas de utilización de Equipos de Computo

A finales de la década de los 60's y principios de los 70's la compañía DEC penetra al mercado con dos elementos primordiales: la fabricación de equipos de menor tamaño y regular capacidad, a los que se denominó minicomputadoras, y el establecimiento de comunicación relativamente confiable entre ellos.

Hacia la mitad de la década de los 70's la delicada tecnología del silicio y de la integración en miniaturas permitió a los fabricantes de computadoras construir mayor inteligencia en máquinas más pequeñas. Estas máquinas llamadas micro computadoras, descongestionaron a las viejas máquinas centrales. A partir de ese momento cada usuario tenía su propia micro computadora en su escritorio.

A esta época se le podría denominar la era del *disco flexible*. Sin embargo, de alguna manera se había retrocedido en la forma de procesar la información, porque nuevamente había que acarrear la información almacenada en los *discos flexibles* de una micro a otra y la poca capacidad estos hacía difícil el manejo de grandes cantidades de datos.

Con la llegada de la *tecnología Winchester* se lograron dispositivos que permitan enorme almacenamiento de información. Una desventaja de esta tecnología era el alto costo que significaba la adquisición de un disco duro. Además, los usuarios tenían la necesidad de compartir información y programas en forma simultánea. Estas razones principalmente, unidas a otras como poder compartir recursos de relativa baja utilización y alto costo, llevo a desarrollar la idea que hoy conocemos como: Redes de Area local (LAN).

1.2 COMPUTO CLIENTE/SERVIDOR.

Actualmente las redes de área local (LAN) son comúnmente implementadas usando el modelo cliente/servidor ya que este es manejado desde cualquier computadora de escritorio, donde el empleado tiene un panorama dentro de la empresa.

A continuación se describen los componentes del modelo cliente /servidor bajo una LAN:

Cliente: Es la combinación de software y hardware, llamada estación de trabajo para un usuario individual, que invoca los servicios de uno o varios servidores, e incluso de otro cliente. El método más común por el cual el cliente solicita los servicios a un servidor es por medio de RPCs (llamada a un Procedimiento Remoto

Servidor: Es la conjunción de software y hardware que responde a los requerimientos de los clientes. Una máquina que provee al cliente manejadores de disco enormes, bases de datos y conectividad a la red. Los servidores pueden ser mainframes, minicomputadoras, grandes estaciones de trabajo o dispositivos de LAN. Más de un servidor puede dar servicios a los clientes.

Red: La red de cómputo es el hardware y software de comunicaciones que enlaza a los clientes con los servidores, de ahí su importancia en la arquitectura cliente/servidor. Dependiendo de su cobertura, las redes se clasifican en redes de área local (LAN), redes de área metropolitana (MAN) y redes de área amplia (WAN). Difieren principalmente en la distancia que pueden cubrir, la tecnología de comunicaciones que emplean, el tipo de equipo, los canales de comunicación que pueden usar y la velocidad a la que pueden operar.

La arquitectura cliente/servidor es un ambiente que reparte apropiadamente los procesos de aplicación entre el cliente y el servidor. El ambiente es heterogéneo comúnmente. La plataforma de hardware, el sistema operativo del cliente y servidor no son el mismo usualmente. Ellos se comunican a través de un bien definido conjunto de *Interfaces de Programas de Aplicación (API's)* y llamadas a *Procedimiento Remoto (RPC's)* estándar.

El diagrama del modelo cliente/servidor es representado en la figura 1.2. El usuario del cliente depende de una estación de trabajo para todas sus necesidades de computo. Si las aplicaciones corren totalmente en la estación de trabajo o usa servicios proporcionados por uno o más servidores es irrelevante.

La arquitectura cliente/servidor es fundamentalmente una plataforma independiente. Los usuarios de una aplicación quieren la funcionalidad que esta proporciona. Los cambios en la plataforma y tecnología básica deberá ser transparente a los usuarios. Costos de capacitación, retraso en los procesos y errores, frustración en la agrupación y cambios bruscos resultan de cambios en un ambiente en el cual los usuarios son sensibles a la tecnología de la plataforma.

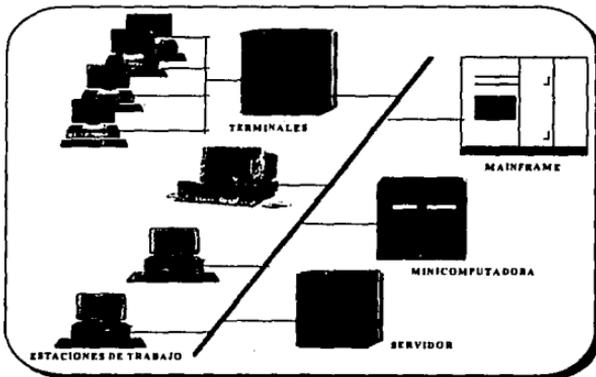


Figura 1.2 Un sistema moderno en Arquitectura Cliente/Servidor

Si los desarrolladores no están conscientes de la plataforma donde desarrollan, las aplicaciones estarán limitadas sólo a la plataforma que ellos conocen.

1.3 VENTAJAS DEL COMPUTO CLIENTE/SERVIDOR.

El modelo cliente/servidor proporciona los medios para integrar aplicaciones a la productividad del personal, tanto para un empleado individual como para un administrador con necesidades específicas de procesamiento de datos, satisfaciendo en forma total los requerimientos de procesamiento de información para toda la empresa.

AMPLIACION DE DATOS COMPARTIDOS

Los datos que son colectados como parte normal del proceso del negocio y almacenados en un servidor están inmediatamente disponibles a todos los usuarios autorizados. El uso de el *Lenguaje de Consulta Estructurado* (SQL, Leguaje de programación para bases de datos) es para definir y manipular los datos proporcionando el soporte de un acceso abierto desde todos los clientes. SQL concede a todos los usuarios autorizados acceso a la información a través de una presentación que es consistente con sus necesidades. Los servicios de red transparentes aseguran que los mismos datos estén disponibles con la misma vigencia a todos los usuarios designados.

SERVICIOS INTEGRADOS

En el modelo cliente/servidor, toda la información que el cliente (usuario) tiene derecho a usar esta disponible en su estación de trabajo. Las herramientas de la terminal correo electrónico, hoja de calculo, presentadores de gráfica y procesador de texto están disponibles y pueden usarse para encargarse de la información proporcionada por aplicaciones y servidores de bases de datos residentes en la red. La figura 1.3 muestra un típico ejemplo de esta integración. Un procesador de texto que incluye entradas desde un paquete de dibujo y hoja de calculo puede ser usado. Las facilidades del intercambio dinámico de datos (DDE) habilita gráficas y hojas de calculo para ser insertadas dentro de un documento del procesador de texto.

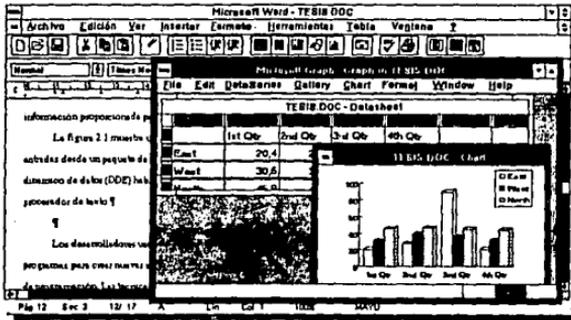


Figura 1.3 Productividad personal y Aplicaciones Integradas

Los desarrolladores usan estas mismas capacidades de manipulación de objetos bajo el control de programas para crear nuevas aplicaciones en una fracción del tiempo consumido por métodos tradicionales de programación. Las técnicas de desarrollo orientadas a objetos están incrementando dramáticamente las facilidades disponibles para los no programadores y usuarios profesionales para construir aplicaciones extendidas.

Otro excelente ejemplo de la integración posible bajo el modelo cliente/servidor es implementado en una estación de servicio para automóviles. En la figura 1.4 se ilustra el conjunto de funciones requeridas por una

estación de servicio de gasolina. El proyecto de automatización de la estación de servicio integra los servicios de medir el flujo de gasolina, el costo de la gasolina despachada, validación de la tarjeta de crédito, administración de la caja registradora, punto de venta, control de inventario, registro de clientes, signos de precios electrónico, monitoreo de tanques, contabilidad, despacho de camiones y otras funciones.

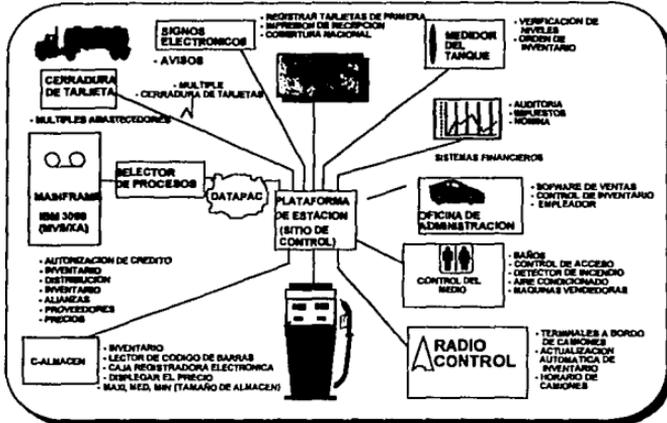


Figura 1.4 Arquitectura de un Sistema Integrado en una Estación de Servicio

COMPARTIR RECURSOS ENTRE DIFERENTES PLATAFORMAS.

El modelo cliente/servidor proporciona oportunidades de llevar a cabo un verdadero sistema de computo abierto, donde no se consideran las plataformas de hardware o las características técnicas del software. De esta forma, los usuarios obtienen los servicios del cliente y accesan bases de datos, comunicaciones y aplicaciones de los servidores. Los sistemas operativos y plataforma de hardware son independientes de la aplicación y enmascarados por las herramientas de desarrollo usadas para construir las aplicaciones

En esta aproximación, las aplicaciones son desarrolladas para encargarse de procesos de la empresa invocados por la existencia de un "evento" creado por el usuario. Un evento es tal como presionar un botón, selección del elemento de una lista, entrar en un cuadro de diálogo, leer un código de barra o el medir el flujo de gasolina ocurren sin que la lógica de la aplicación sea sensible a la plataforma física. Todo lo anterior se logra gracias al uso de interfaces estándar.

INTERCAMBIABILIDAD E INTEROPERABILIDAD DE DATOS

SQL es un estándar que permite la definición de datos y lenguaje de acceso. Este estándar ha permitido desarrollar motores de bases de datos que administran los datos como tablas de SQL. La mayoría de herramientas de desarrollo usadas en arquitecturas cliente/servidor son back-end en servidores de base de datos

accesados a través de SQL y frond-end en las clientes. Los servicios de red proporcionan conectividad transparente entre el cliente y un servidor local o remoto.

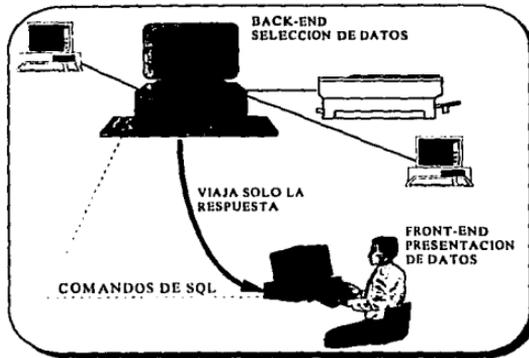


Figura 1.5 Back-End y Front-End

La sintaxis de SQL no especifica una asignación o plataforma. Esta transparencia habilita las tablas para ser movidas entre diversas plataformas sin afectar el código de la aplicación.

ACCESO FISICO A DATOS ENMASCARADOS

El uso de nuevos tipos de datos como objetos binarios extensos (BLOB's) habilitan otro tipo de información como imágenes, vídeo y audio para ser almacenados y accesados usando las mismas declaraciones SQL para su acceso. Las RPC' s frecuentemente incluyen facilidades de conversión de datos para trasladar los datos almacenados en un procesador a otro con un formato aceptable.

LOCALIZACION INDEPENDIENTE DE DATOS Y PROCESAMIENTO

A los usuarios que trabajan en una aplicación desde el escritorio no les concierne la localización o tecnología de la aplicación involucrada.

La figura 1.6 ilustra la evolución de la visión de los usuarios respecto a la plataforma de computo. En los 70's, los usuarios trabajando con un mainframe IBM, una minicomputadora VAX o una de las primeras aplicaciones de micro computadora, estaban conscientes de que plataforma usaban. Cada plataforma requería de una secuencia única de parámetros de seguridad, opciones de teclado, ayuda al usuario, navegación y recuperación de errores. En el actual mundo centralizado al usuario, la estación de trabajo provee el punto de acceso al grupo de trabajo y servicios de la empresa sin tomar en cuenta la plataforma o la ejecución de la aplicación. Los servicios estándar se suministran consistentemente entre todas las aplicaciones. La red puede estar basada en OSI, TCP/IP o SNA pero no cambian los requerimientos lógicos de codificación.

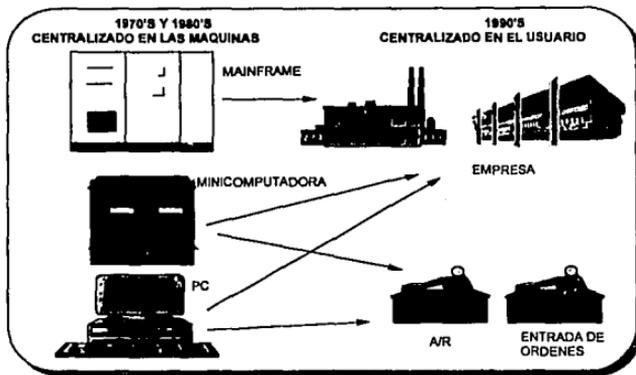


Figura 1.6 La transformación de la Computación

1.4 DESVENTAJAS DEL COMPUTO CLIENTE/SERVIDOR

Las aplicaciones cliente/servidor suelen ser más complejas que las tradicionales en la forma de anfitrión/terminales, y que exigen más de la red.

Costo elevado.

La seguridad informática mantienen, que bajo cliente/servidor es más sencilla de romper. Pero la respuesta a esta aseveración es que la seguridad está sujeta al diseño y medidas de seguridad que se incorporen al ambiente, y a las aplicaciones desarrolladas para operar bajo la arquitectura cliente/servidor.

1.5 EL MODELO OSI

En 1978 la Internacional Standards Organization (ISO) propuso un modelo para comunicaciones de redes locales al que titularon: OSI, modelo de referencia de interconexión de sistemas abiertos (reference Modelo of Open Systems Interconnection). "Interconexión de Sistemas Abiertos" significa el intercambio de información, entre terminales, ordenadores, personas, redes o procesos.

El modelo define donde se han de efectuar las tareas, pero no como se han de efectuar. No especifica servicios ni protocolos. El modelo OSI intenta proporcionar una base común para coordinar el desarrollo de estándar dirigidos a la conexión entre sistemas.

El sistema se compone de un conjunto de ordenadores de subsistemas o niveles. Los niveles del modelo OSI están separados por interfaces. Los niveles adyacentes se comunican entre sí por medio de un interface común.

Los objetivos primordiales de este modelo son:

1. Proporcionar una serie de normas para la comunicación entre sistemas.
2. Eliminar todos los impedimentos técnicos que pudieran existir para la comunicación entre sistemas.

3. Abastecer el funcionamiento interno de los sistemas individuales.
4. Definir los puntos de interconexión para el intercambio de información entre los sistemas.
5. Limitar el número de opciones, para incrementar el número de posibilidades de comunicación sin necesidad de numerosas conversiones y traducciones entre diferentes productos.
6. Ofrecer un punto de partida válido desde el que comienza en caso de que las normas del estándar no satisfaga todas las necesidades.

1.5.1. LOS 7 NIVELES DEL MODELO OSI

NIVEL 1: FISICO

El nivel físico define las características eléctricas y mecánicas de las interfaces de la red necesarios para establecer y mantener la conexión física. Este nivel está pensado para atender una gran variedad de medios físicos y procedimiento de control. También define las comunicaciones por radio y por infrarrojos. El nivel físico especifica lo siguiente:

- Conexiones eléctricas y características mecánicas.
- Como se convierte un flujo de bits la información que ha sido empaquetada (metida en sobre o tramas), para ser transmitida por cable
- Como consigue el acceso a la Tarjeta de red.

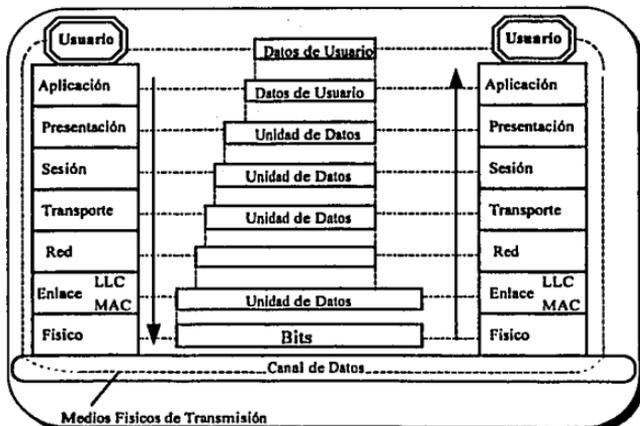


Figura 1.7 Diagrama del modelo OSI

NIVEL 2: ENLACE DE DATOS

El nivel de enlace de datos establece y mantiene comunicaciones entre los usuarios. Es el responsable de mantener un canal sin errores, detectando y corrigiendo los que se puedan producir.

Los protocolos relacionados con este nivel (IEEE 802), son los encargados del formato de los bloques de datos, de los códigos de dirección, de la detección y recuperación de errores y del orden de los datos transmitidos.

El nivel de enlace de datos se divide en 2 subniveles.

- El nivel de enlace lógico (LLC): recibe paquetes de niveles superiores y los envía al nivel MAC.
- El nivel de Control de acceso al medio (MAC): se encarga de enviar los paquetes a sus destinos.

El subnivel MAC se sitúa por debajo del LLC para conseguir que la comunicación extremo a extremo sea independiente del método de acceso utilizado en la red.

NIVEL 3: NIVEL DE RED

El nivel de red define protocolos para abrir y mantener un camino entre equipos de la red. Se ocupa del modo en que se mueven los paquetes. El nivel de red puede mirar la información sobre la dirección y determinar el mejor camino para transmitir los datos a su destino. Esto es importante en las interconexiones entre redes, donde se enlazan varios segmentos LAN. Si un paquete lleva la dirección de una estación local, se envía directamente allí. Si lleva la dirección de otro segmento en la red, el paquete es enviado a un dispositivo de encadenamiento (routing), que envía el paquete usando el mejor camino a través de los routers.

Los protocolos relacionados con este nivel se encargan de la administración y gestión de los datos, emisión de mensajes de estado, regularización del tráfico de la red y reparto de trabajo entre las distintas unidades de interfaz y la estación central. Se encargan también de que todos los paquetes lleguen correctamente a su destino.

NIVEL 4: NIVEL DE TRANSPORTE:

Es el encargado de la transferencia de los datos entre el emisor y el receptor y de mantener el flujo de la red. Su función básica es aceptar datos del nivel de sesión, dividirlos en mensajes y pasar estos al nivel de red. Comprueba también si los mensajes llegan correctamente a su destino, si falta información del paquete al acabar la recepción, el protocolo de nivel de transporte acuerda un reenvío con el nivel de transporte del equipo que mando los paquetes. Este nivel se asegura de que toda la información se reciba en el orden adecuado, representando la línea de separación entre la transmisión de los datos y el proceso de los mismos. Los protocolos de este nivel controlan la distribución de los mensajes y evitan que se pierdan o dupliquen mensajes.

Esta diseñado para mantener al usuario al margen de algunos de los aspectos físicos y funcionales de la red de paquetes. Se encarga además de la facturación entre los dos extremos.

NIVEL 5: NIVEL DE SESION

– Organiza, sincroniza y se encarga del diálogo entre los usuarios; es decir, es la interfaz entre el usuario y la red. El usuario ha de dirigirse a este nivel para establecer una conexión con otra estación, una vez hecha la conexión, el nivel de sesión sincroniza el diálogo y se encarga del intercambio de datos.

Los protocolos de este nivel incluyen reglas para establecer y dar por finalizadas las conexiones, verificando al mismo tiempo si esta teniendo lugar la comunicación adecuada y comunicando la red con el sistema operativo. Cada usuario puede seleccionar el tipo de control y de sincronización que desea de la red, como por ejemplo:

- Diálogo bidireccional alternado o bidireccional simultáneo.
- Puntos de sincronización para comprobaciones intermedias y recuperaciones durante la transferencia de ficheros.
- Abortos y reanques.
- Flujo de datos normal y acelerados.

El nivel de sesión posee una serie de servicios específicos, primitivas y unidades del protocolo de datos, que están definidos en los documentos de ISO y CCCITT.

NIVEL 6: NIVEL DE PRESENTACION

Traduce la información del formato de máquinas a un formato que pueda entender el usuario. Una de las funciones mas importantes es la traducción de los distintos formatos de fichero y de terminal, y de los diferentes sistemas de decodificación (por ejemplo, de ASCII a EBCDIC).

Determina la forma de representación de los datos según este modelo, sin preocuparse por el significado o semántica. En realidad sus funciones son bastantes limitadas. Este nivel consta de muchas tablas sintácticas (correspondiente a códigos como el teletipo, ASCII, Videotex, etc.). A si mismo es capaz de crear visualizaciones de terminales virtuales (terminales de paso), también puede resolver la recepción de un mensaje electrónico procedente del nivel de aplicación y encargar al nivel del otro extremo que proporcione al otro nivel de aplicación un formato de página determinado (por ejemplo, una composición tipográfica).

NIVEL 7: NIVEL DE APLICACION

En este nivel, el sistema operativo de red y sus aplicaciones se hacen disponibles a los usuarios. Los usuarios emiten ordenes para requerir los servicios de red, y esas ordenes son paquetizadas y enviadas por la red a través de los niveles de protocolos mas bajos. Los protocolos de este nivel se ocupan del soporte de los programas de aplicación, como claves de acceso, transferencia de ficheros, etc.

Se encarga de atender al proceso de aplicación del usuario final. A diferencia del nivel de presentación, este nivel tiene en cuenta la semántica de los datos. Contiene varios elementos de servicio capaces de gestionar

procesos de aplicación tales como la gestión de trabajos, el intercambio de cambios financieros, sentencias de enviar y recibir de distintos lenguajes de programación y el intercambio de datos comerciales. Asimismo, este nivel maneja los conceptos de terminal virtual y fichero virtual.

1.5.2. FUNCIONAMIENTO DEL MODELO OSI

Todos los niveles de la estructura disponen de un conjunto de servicios para el nivel superior e inferior. La relación entre los distintos niveles y la información que se ha de pasar están claramente definidas.

Los requerimientos específicos y los servicios difieren de nivel a nivel pero los conceptos son similares. Esto nos lleva aun concepto de unificación, en que el nivel N de la arquitectura proporciona un servicio al nivel N+1 arriba de él. Por tanto la entidad N es la proveedora del servicio a la entidad N+1 usuaria del servicio.

Los puntos de acceso al servicio deben establecer conexiones entre entidades a mismo nivel. Por tanto, como uno de sus servicios, el nivel N establece conexiones entre puntos de acceso al servicio N.

Los datos que se transfieren de unidades iguales contienen datos del usuario que pasan del nivel superior, e información de control de protocolo que se añade al nivel en cuestión (fig. 1.8). UDP significa unidades de datos de protocolo, el bloque de datos que contiene información de protocolos ICP, añadida en nivel en cuestión, como se muestra, y datos del usuario, que se originan en la capa superior. La UDP (N+1), al cruzar la interface entre los niveles (N+1) y N, se hace corresponder con la unidad de datos del servicio N, UDS N como se muestra.

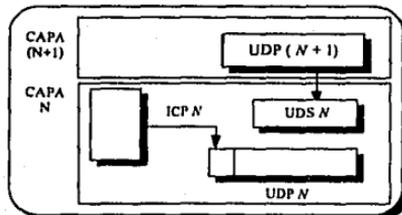


Figura 1.8 Unidades de Datos en el Sistema OSI

La longitud de la UDP N transmitida entre las entidades de categoría N se negociaría durante la fase de conexión. Cuando una UDP N llega a su destino, la entidad de categoría N en el destino, se procede a quitarle el encabezamiento, el ICP N, antes de enviarse al nivel (N+1) del destino. También se podría invertir la segmentación o concatenación.

Además de las unidades de datos del protocolo N que se intercambia entre unidades de la misma categoría en el nivel N se intercambia información de control de interface N entre una entidad (N+1) y una entidad N en el nivel inferior para coordinar su operación. El control de flujo de las unidades de datos se puede ejercer entre entidades de la misma categoría en el mismo nivel, a si como entre entidades (N+1) y N a través de la interfaz en el punto de acceso al servicio.

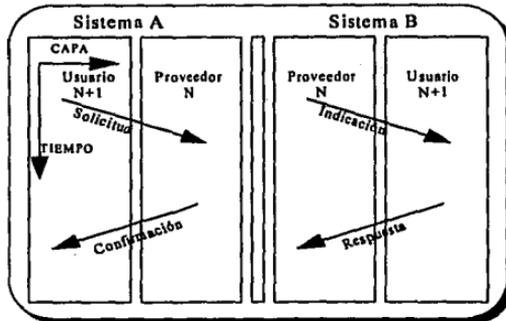


Figura 1.9 Cuatro primitivas básicas de la Arquitectura OSI

OSI ha establecido estándares sobre el uso de 4 primitivas básicas de servicio a cada nivel de la arquitectura para proporcionar la interacción entre el usuario del servicio en un nivel y el proveedor del servicio en el nivel inferior. Estas primitivas de servicio ofrecen los elementos básicos para definir un intercambio entre usuarios del servicio; los cuatro tipos son: Solicitud, Indicación, Respuesta y Confirmación. No necesariamente se usan los 4 tipos en los diferentes niveles; no se pueden usar todos en las diferentes fases de un intercambio entre entidades iguales en un nivel dado. Las 4 primitivas se presenta de modo esquemático en la fig. 1.9.

Nótese en la figura que la solicitud es emitida por un usuario del servicio en un nivel (N+1) dado en el sistema A para invocar un procedimiento del protocolo del servicio del proveedor en el nivel N inferior, lo que da como resultado un mensaje de nivel N, que se envía como una UDP N (o conjunto de UDP N) al sistema B. La recepción de la UDP N en el nivel N del sistema B a veces causa la emisión de una primitiva de indicación por parte del proveedor del servicio en ese nivel. Esta primitiva de indicación se emite en cualquiera de estos dos casos: puede indicar que el usuario del servicio en B está a punto de invocar un procedimiento del protocolo en su nivel (N+1) o que el proveedor del servicio en el nivel N invocó un procedimiento en el punto de acceso del servicio de la categoría. En resumen, la primitiva de indicación en el sistema B se emite, en este ejemplo, como respuesta a la primitiva de solicitud emitida antes en el sistema A. Como se muestra en la figura, el proveedor del servicio emite la primitiva de respuesta en el nivel N+1 del sistema B como respuesta a la indicación que aparece antes en el punto de acceso al servicio entre los niveles N y (N+1) del sistema B. Esta primitiva de respuesta es una directiva al protocolo del nivel N para completar el procedimiento invocado previamente por la primitiva de indicación. El protocolo en el nivel N genera una unidad de datos del protocolo que se transmite a través de la red, aparece en el nivel N del sistema A y resulta en una primitiva de confirmación emitida por el proveedor del servicio en el sistema A. Esto lleva a la culminación, en el punto de acceso al servicio entre N y (N+1) en el sistema A, del procedimiento evocado previamente por la solicitud en ese punto.

1.6. DOWNSIZE.

Introducción

Downsizing. Migración de aplicaciones de misión crítica a servidores potentes que ofrecen a los clientes acceso transparente a estas aplicaciones a través del sistema empresarial extendido. Reducción de tamaño. Implica un proceso de reducción de grandes mainframes a minicomputadoras. Este concepto también puede aplicarse a reducción de proyectos que impliquen menor personal.

Las grandes corporaciones, enfrentan el incremento de mercados competitivos que requieren inmediata respuesta a las necesidades de los clientes, ellos han cambiado al downsizing para llegar a ser más eficientes.

En cambio, la definición actualizada de downsizing describe el proceso de aplicaciones de computadora de misión crítica, tales como base de datos de clientes y software de control de manufactura, "bajando" de un mainframe o minicomputadora a computadoras personales que están conectadas vía LAN's.

Esta clase de downsizing trae consigo ganancia en productividad sin efectos negativos en el tamaño de la fuerza de trabajo de una organización, como lo haría la definición original de downsizing.

El downsizing en ambas de sus formas, emplean la suspensión, uno especialmente dentro de la administración media y otro moviendo capacidades de computación crítica lejos del anfitrión centralizado, para llevar a incrementar los beneficios. De cualquier forma, la versión de los despidos se hace a expensas de los recursos humanos, mientras que la versión de computadora toma ventaja de mejorar la tecnología de redes para agregar eficiencia.

1.6.1. INCREMENTO DE PRODUCTIVIDAD

El downsizing en los recursos computacionales de una compañía puede producir ganancia en productividad. En el tradicional medio ambiente basado en mainframe, el MIS (Manager Information System), departamento que controla el acceso a los recursos computacionales, específicamente el mainframe o minicomputadora y los datos almacenados en ellas. Donde, sólo los empleados del MIS tienen acceso directo a el mainframe y su cuarto de almacenamiento de información valiosa. Un número de factores pueden necesitarse para semejante afirmación. Primero, las aplicaciones del mainframe son difíciles de aprender y correr, y poca gente tiene tiempo o inclinación de llegar a sacar provecho en la manipulación de datos en el mainframe. Problemas de seguridad, también, prohíben amplio acceso a el sistema.

En semejante situación, los empleados de otros departamentos requieren de reportes de MIS, el cual actúa como un intermediario entre empleados y los datos que ellos necesitan para hacer sus trabajos. Si el departamento de MIS es mal empleado, puede tomar varios días o mas llevar un reporte a los empleados que lo piden. Porque los retrasos pueden representar la diferencia en mantener clientes satisfechos, guardando datos sólo en mainframes no es una forma particularmente eficiente de tratar con problemas de tiempo crítico, todavía piensan muchas organizaciones grandes depender de mainframes para sus recursos de computo.

Moviendo los datos de la compañía desde un mainframe a aplicaciones basadas en LAN eliminan los intermediarios de MIS. El downsizing en recursos de cómputo hace una línea frontal de empleados más productivos dándoles acceso directo a los datos cruciales y aplicaciones que necesitan. El acceso directo a información pertinente los habilita para tomar decisiones más rápidamente.

1.6.2. EL MODELO CLIENTE/SERVIDOR EN LAS ORGANIZACIONES

En un mundo competitivo se hace necesario para las organizaciones tomar ventaja de cada oportunidad para reducir costos, mejorar su calidad y proporcionar servicio. La mayoría de las organizaciones hoy reconocen la necesidad de guiar el mercado, para ser competitivos y demostrar valor agregado.

Una estrategia que están adoptando muchas organizaciones es adelgazar la jerarquía administrativa. Con la eliminación de capas de la administración media, los individuos restantes deben ser autorizados para hacer de la estrategia un éxito. La información necesaria para la toma racional de decisiones debe estar disponible a estos individuos. La tecnología de la información (IT) es un vehículo efectivo para soportar la implementación de esta estrategia; frecuentemente esta no es usada efectivamente. El modelo cliente/servidor proporciona poder al escritorio, con información disponible para soportar el proceso de toma de decisiones y habilitar la autoridad para tomar decisiones.

El grupo Gartner, un equipo de analistas en la industria del cómputo, noto un amplio abismo entre la expectativas de los usuarios y la capacidad de los sistemas de información de las organizaciones para llenarlo. El hueco ha sido alimentado con el dramático incremento de usuarios finales conformes con tecnología (principalmente por el común alfabetismo en computación); continua baja en costos en tecnología de hardware esencial; intensificación en extremo de las promesas en la publicidad por parte de vendedores; incremento en el tiempo de retraso entre las fechas de salida fijadas por los vendedores y la liberación final de un producto (conocido como "vaporware"); y surgimiento de las interfaces de usuario gráfica (GUI) como la solución percibida a todos los problemas de cómputo.

La computación cliente/servidor es la tecnología capaz de cerrar este abismo. Esta tecnología, particularmente cuando esta integrada dentro del proceso normal de los negocios, se puede tomar ventaja de este nuevo alfabetismo, tecnología con costos efectivo y GUI amigables. En conjunción con un medio ambiente de desarrollo de sistemas (SDE) bien construido, la posibilidad de la computación cliente/servidor de usar la tecnología de hoy y estar en posición de llevar ventaja de las promesas de vendedores para llegar a hacerlas realidad.

CAPITULO II

ELEMENTOS Y

ESTRUCTURAS

DE REDES

LAN

CAPITULO II: ELEMENTOS Y ESTRUCTURAS DE REDES LAN

Uno de los conceptos mas importantes dentro de las redes es la conectividad, ya que esta tiene la capacidad de comunicar computadoras entre si. Una red se puede conceptualizar como un tendido de cable al que se conectan diferentes máquinas por medio de Tarjetas de comunicación. El cable constituye la vía de comunicación entre las máquinas, es decir, el medio por el que trasmite la información.

Existen diferentes tipos de redes de acuerdo al propósito perseguido en el trabajo, sin embargo una de las funciones primordiales en todas las redes es el compartimiento de recursos. Se entiende a recursos de red a todos los dispositivos, ya sea de tipo software como pueden ser sistemas operativos, lenguajes de programación, paquetes de aplicaciones y documentos, o bien de tipo hardware entre los que se encuentran las unidades de almacenamiento (discos duros, CD-ROMS, cintas magnéticas, etc.), impresoras, gráficos, digitalizadores de imagen y modems, entre otros.

2.1 MEDIOS DE COMUNICACION

Cada esquema de cableado de red exige un tipo de cable específico. La distancia entre las estaciones de trabajo de la red afecta al tipo de cable que se puede elegir y a la cantidad de cable que se necesita. Las señales que van por el cable pueden producir radiaciones con armónicos que interfieren con equipos externos.

Aun siendo el cable correcto para cada tipo de red puede haber problemas si pasa próximo a luces fluorescentes o fuentes de potencia por los techos o paredes. Este tipo de interferencia no tiene importancia en la transmisión de la voz, pero si para la transmisión de datos. Los tipos de cable mas usados son:

2.1.1 CABLE COAXIAL

Este se forma por un alambre conductor básico cubierto por una placa metálica que actúa como tierra. El alambre conductor y la tierra se encuentra separados por un aislante plástico y finalmente todo el conjunto esta cubierto por una capa exterior también aislante a la que por común se le llama jacket.

Los cables coaxiales pueden ser de diferentes tipos y anchos, sin embargo su principal característica es que pueden transportar una señal eléctrica a mayor distancia entre mas grueso es el conductor. El cable grueso suele ser mas caro y menos flexible. Por tal razón cuando tiene que colocarse en instalaciones donde ya existen canales para cableado o conductos con espacio reducido y, sobre todo, limitado en las esquinas o dobles resulta mas conveniente utilizar el cable delgado, debido a que las instalaciones de ductos para cable por lo general son muy costosas, esto puede ser un factor determinante para implantación de una red local.

La alternativa de colocar cables coaxiales en redes locales tiene una relación de costo beneficio muy buena.

Las principales ventajas des esta cables son:

- Transmisión de voz, videos y datos.
- Fácil instalación
- Compatibilidad con Arcnet y Ethernet.
- Ancho de banda de 10 Mbps
- Distancia hasta 600 metros sin necesidad de repetidores.
- Muy buena tolerancia a interferencia debida a factores ambientales.
- Costo relativamente bajo.

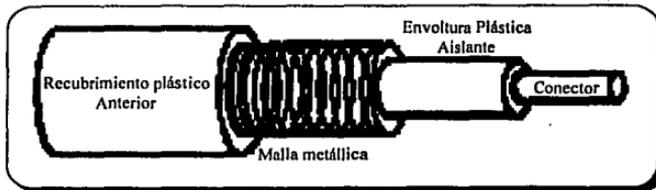


Figura. 2.1 Cable Coaxial

2.1.2. CABLE TELEFONICO

Este es formado principalmente de dos alambres de cobre que se encuentran aislados por una cubierta plástica y torcidos uno contra el otro. Con esta característica se distingue con el nombre de cables de par torcido (Twisted Pair). A su vez, se encuentra cubierto por una cubierta aislante y protectora de la capa exterior denominada Jacket.

Los cables con los conductores de cobre mas delgado y menos protegido por un jacket están dentro de la clasificación de cable tipo UTP (Unshielded Twisted Pair / Par torcido sin Blindar), son sumamente baratos, flexibles y permiten manipular una señal a una distancia máxima de 110 metros sin el uso de amplificadores.

Los cables de conductores más gruesos y muy bien cubiertos por un jacket son denominados de tipo STP (Shielded Twisted Pair / Par torcido Blindado). Estos son muy caros y menos flexibles que los UTP, pero permiten un rango de operación de hasta 500 metros.

En general el cable telefónico viene en conjuntos típicos de 2, 3, 4, 6, 12, 16 y 25 pares de cables torcido, sin embargo para redes locales de tipo UTP solo se necesitan 2 pares de cables para conectar a cada nodo de la red. Las causas de falla generalmente se deben a factores humanos (una ruptura accidental) y rara vez a factores ambientales, debido a que la vida útil de un cable bien instalado y protegido supera los 10 años.

Las principales ventajas de los cables telefónicos son:

- Tecnología conocida.
- Facilidad y rapidez de instalación
- Compatibilidad con Ethernet y Token Ring.

- Ancho de banda de 10 Mbps
- Distancia de hasta 110 metros con cables UTP Y 500 metros con cable STP.
- Buena tolerancia a interferencia debido a factores ambientales.

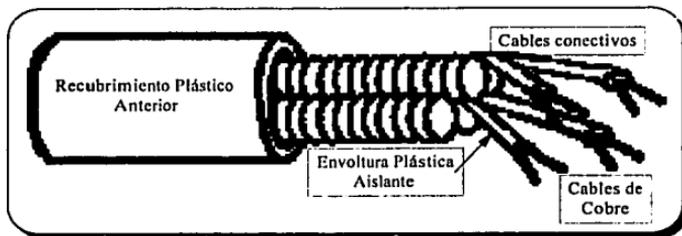


Figura 2.2 Cable Telefónico

2.1.3. FIBRA OPTICA:

Este tipo de cable es empleado generalmente donde son grandes las distancias, donde se requiere de una alta capacidad de aplicación de comunicación y donde el ruido a cualquier tipo de interferencia es un factor a considerar.

El cable de Fibra Optica se compone de una fibra muy delgada elaborada de dos tipos de vidrio con diferentes índices de refracción, uno para la parte interior y otro para la parte exterior. Esta diferencia en la refracción previene que la luz penetre en una parte de la fibra óptica hasta la parte exterior, evitando así la pérdida de la información. La fibra óptica a su vez se encuentra cubierta por una placa aislante y protectora en la parte más exterior para darle mayor integridad estructural al cable. Sin embargo es, extremadamente flexible ya que se pueden realizar giros de hasta 360 grados sin problemas de afectación en el cable. El diámetro de la fibra interior más comúnmente usado es de 62.5 micras y el de la fibra exterior, de 125. Presentan una atenuación máxima de 4 dB/Km.

Para información de transmisión de redes locales vía fibra óptica se utiliza una fibra como transmisor y otra como receptor, es por esto que generalmente se producen en conjunto de mínimo dos fibras por cable. La distancia máximas obtenidas para redes locales son de 2000 metros de nodo a nodo sin el uso de amplificadores.

Entre las principales ventajas de fibra óptica se encuentran las siguientes:

- Transmisión de voz, vídeo y datos por el mismo canal.
- Aplicaciones de alta velocidad.
- No genera señales eléctricas o magnéticas.
- Inmune a interferencias y relámpagos.
- Puede propagar una señal sin necesidad de utilizar un amplificador a distancias de 2000 metros.
- Tiene un ancho de banda de 200 Mbps.
- Excelente tolerancia a factores ambientales.

- Ofrece la mayor capacidad de adaptación a nuevas normas de rendimiento.
- Compatibilidad con Ethernet, Token Ring y Interface de datos distribuidos por fibra (FDDI Fiber Data Distributed Fiber)



Figura 2.3 Cable de Fibra Óptica

2.2 TOPOLOGIA DE REDES

La Topología es la forma o conectividad de la red. Las tarjetas de comunicación determinan la topología o disposición física del cableado de la red, es decir, la forma del tendido de cable al que se conectan las máquinas y, por tanto la forma en que viaja la información entre ellas.

Al elegir la topología de la red se deberá de tener en cuenta las siguientes consideraciones:

1. Proporcionar la máxima fiabilidad posible para garantizar la recepción correcta de todo el tráfico.
2. Economizar el tráfico entre las estaciones de trabajo transmisor y el receptor a través del camino más económico dentro de la red (aunque si se consideran mas importantes otros factores como la fiabilidad, este camino de coste mínimo puede no ser el más conveniente).
3. Proporcionar al usuario final un tiempo de respuesta óptimo y un caudal eficaz máximo.

Existen 3 Topologías primordiales:

2.2.1. BUS LINEAL

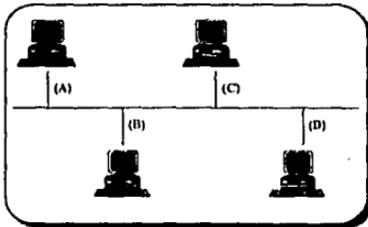


Figura. 2.4 Topología de Bus

Esta topología conecta las microcomputadoras en una sola línea de cable todas las máquinas usan dicha línea para transferir información a través de la red. Las máquinas podrán enviar o recibir información en dos direcciones. Cada punto de la red "escuchará" la información que transmite por el cable. Cuando una máquina reconozca su dirección de identificación aceptará la información.

En los extremos del bus o tronco del cableado se instalan terminadores (resistencias eléctricas), cuyo cometido es extinguir las señales que viajan por el cable para así evitar reflejos que puedan perturbar las señales

originales. Las tarjetas de red de tecnología "Ethernet" utiliza esta topología. La principal limitación de esta topología esta en el hecho de que suele existir un sólo canal de comunicación para todos los dispositivos de la red. En consecuencia, si el canal de comunicación falla toda la red deja de funcionar. Algunos fabricantes proporcionan canales completamente redundantes por si falla el canal principal Otro inconveniente es la dificultad aislar las averías de los componentes individuales conectados al bus. En la figura 2.4 se muestra un diagrama de la topología en bus lineal .

2.2.2. TOPOLOGIA DE ESTRELLA

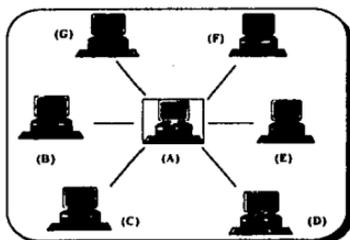


Figura 2.5 Topología de Estrella

Usa un dispositivo central, como un divisor de señales o un centro de cableado que estará conectado directamente por un cable a cada una de las máquinas de la red. Esta se utilizo a lo largo de los 60's y a principios de los 70's, su software no es complicado y su flujo de trafico es sencillo. Las microcomputadoras conectadas con esta topología se comunican a través de transmisores que pasan siempre por el centro de la estrella antes de llegar a su destino.

Al fallar el dispositivo central se tendrá una caída total del sistema. Las tarjetas de red "Archnet" utilizan esta topología. Su capacidad de procesamiento distribuido es limitada. Cada una de las terminales es responsable de encaminar el trafico al resto de los componentes; se encarga de localizar la averías. En otro sistema se estableció redundancia en el nodo central, como medida de seguridad, con lo cual la fiabilidad aumentó considerablemente. Ver figura 2.5

5.2.3. TOPOLOGIA DE ANILLO

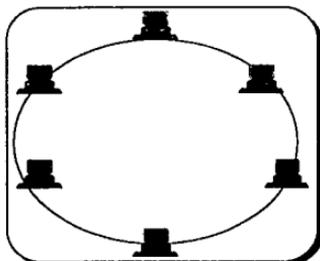


Figura 2.6 Topología de Anillo

En esta las máquinas de la red son conectadas punto a punto para constituir el anillo al conectar la última máquina con la primera, mostrada en la figura 2.6. El modo de funcionamiento de una red con esta topología se explica la sección dedica al paso de testigo en anillo ya que este es el protocolo que utiliza. Hoy en día las redes de anillo son controladas por Unidades de Acceso múltiple, dispositivos conectados a todas las máquinas cuyo cometido es dirigir la información a cada punto de la red. El anillo queda en el interior de la UAM.

2.3 TARJETAS DE INTERFACES DE RED

Las tarjetas de red son dispositivos construidos con base de Circuitos Integrados que se instalan en el interior de las máquinas y sirven para codificar la información y transmitirla por el cable de la red a otro punto de esta o bien decodificarla en caso de estar recibiendo un paquete de información. Dependiendo del tipo de tarjeta en uso se determina una disposición física o "Topología" para el cableado de la red.

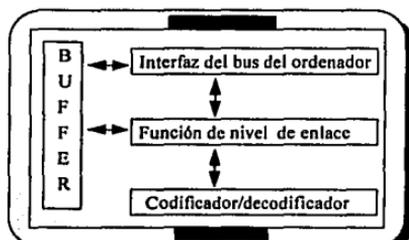


Figura 2.7 Funcionamiento de la Tarjeta de Red

Las tarjetas de Interface permiten empaquetar la información y transmitirla a cierta velocidad y de acuerdo con características determinadas de envía. Estas varían según la topología y el protocolo de red. Todas las máquinas de una red deben de tener el mismo tipo de tarjeta, ya que de ello depende en código de comunicación en la red. En la figura 2.7 se muestra como se conecta un microordenador a la red por medio de una tarjeta de interfaz de red.

A nivel técnico, el interfaz de red se puede dividir en dos partes:

En interfaz de comunicaciones, que contiene las funciones de red, y el interfaz del ordenador, que contiene las funciones específicas del ordenador. En la figura 2.8 se muestran ambas partes.

El interfaz de comunicaciones es la unidad que se comunica lógicamente con la red y lleva acabo todas las funciones de transmisión. Esta interfaz acepta datos de la estación a la que está conectado, retiene los datos hasta que el canal de comunicación está disponible y entonces transmite los datos. El interfaz de comunicaciones controla también si se han recibido mensajes dirigidos a su estación, almacena datos en un Buffer y después transfiere los datos a dispositivo (microordenador, impresora, etc.).

La conexión física entre la estación, sea cual fuere su tipo, y la red se logra por medio de un cable secundario que conecta la interfaz de comunicaciones y el cable principal de la red. Los cables se conectan por medio de un "derivador".

El interfaz del ordenador es el que proporciona la conexión entre los circuitos internos de una determinada estación y la unidad de interfaz de comunicaciones. Esta interfaz esta conectado a una estructura de entrada/salida de la estación y controla el intercambio de datos entre la estación y la interfaz de red.

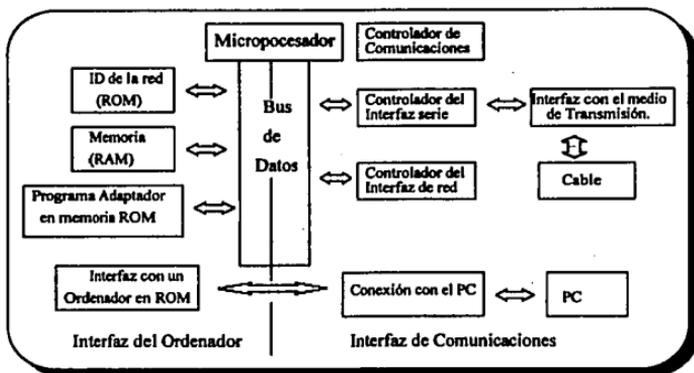


Figura 2.8 Interfaz con el Microprocesador

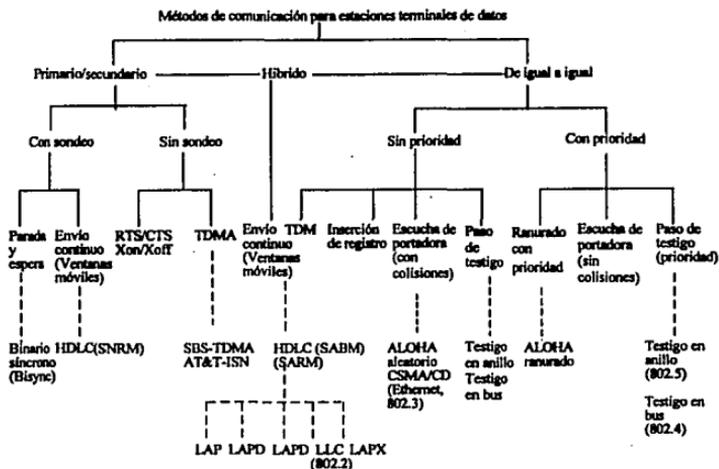
2.4 ESTANDAR IEEE 802, FDDI y X.25

Los protocolos son el software convencional para cada una de las tarjetas utilizadas en redes, utilizado para comunicar dos o más dispositivos de la misma naturaleza.

A continuación se da una descripción general del "árbol de clasificación" de la figura 2.9¹. Las estaciones de trabajo se comunican entre sí mediante las técnicas descritas en la figura 2.9. La mayoría de los protocolos que aparecen en la figura se conocen como protocolos de línea (enlace o canal) o controles de enlace de datos (DLC). Reciben este nombre por que su función es gobernar el flujo de tráfico entre estaciones a través de un canal físico de comunicaciones. En otra sección de este capítulo estudiaremos otros protocolos de alto nivel que ofrecen funciones adicionales a la red.

Un método muy utilizado para gestionar un canal de comunicación es el llamado protocolo *primario/secundario (maestro/esclavo)*. En esta técnica se designa una estación de trabajo, un concentrador o un Ordenador de conmutación como nodo principal del canal. Este nodo primario (por lo general un ordenador) controla todas las demás estaciones y determina si los dispositivos pueden comunicarse y, en caso afirmativo, cuando deben hacerlo. El segundo método importante es el protocolo de *igual a igual*. En este sistema ningún nodo es el principal, y por lo general todos los nodos poseen la misma autoridad sobre el canal, sin tener todos un idéntico acceso a la red, ya que pueden existir prioridades preestablecidas entre los distintos elementos. A pesar de ello, la ausencia de un nodo primario proporciona a todos los nodos las mismas oportunidades de utilizar los recursos de la red. Los sistemas de igual a igual son frecuentes en las redes locales con topologías en anillo, en bus y en malla, así como en determinados sistemas híbridos.

¹Uyless Black., "Redes de computadoras: protocolos, normas e interfaces", Madrid, 1989.



LAP	Procedimiento de acceso de enlace. D Canal D; X Extendido; B Balanceado
SDLC	Control Síncrono de enlace
TDM	Multiplexado por división temporal
HDLC	Control de enlace de alto nivel
TDMA	Acceso múltiple por división temporal
RTS/CTS	Solicitar para enviar/Limpiar para enviar
LLC	Control de enlace lógico
CSMA/CD	Detección de portadora de acceso múltiple con detección de portadora
SBS	Sistema de negocios por satélite

Figura 2.9 Clasificación de redes de computadoras

En la actualidad los protocolos pertenecientes a la clasificación de igual a igual, hablando a nivel de fabricantes y uso en sistemas cliente/servidor, son los más recurridos. Sin querer afirmar con esto que no haya uso en sistemas cliente/servidor de los otros protocolos "igual a igual" mostrados en la figura 2.9.

El Instituto de ingenieros Eléctricos y electrónicos (IEEE) ha establecido seis subcomités con el fin de desarrollar estándares para redes de área local. Todos estos comités reciben la denominación colectiva de Comités de Normalización de redes locales IEEE 802:

- 802.1 Gestión y Niveles superiores (HILI)
- 802.2 Control lógico de enlace (LLC)
- 802.3 Detección de Portadora de Acceso Múltiple con Detección de Portadora (CSMA/CD)
- 802.4 Token Bus (Paso de testigo en bus)
- 802.5 Paso de testigo en anillo (Token Ring)
- 802.6 Redes Metropolitanas (MAN Metropolitan Area Networks)

La explicación se concentrara en los estándares IEEE 802.2, IEEE 802.3 e IEEE 802.5 ya que estos son los principales protocolos usados en la actualidad. Además se ajustan perfectamente al modelo OSI implementándose en la capa de enlace de datos. El IEEE 802.3 Y 802.5 son conocidos como protocolos de Control de Acceso al Medio (MAC). El IEEE 802.2 es utilizado como protocolo para control de enlace lógico (LLC).

FDDI es el mayor avance en lo que respecta a funcionalidad en LAN's. Ha sido desarrollado por el comité X3T9.5 del Instituto Americano de Normalización (ANSI). Siguiendo el concepto estructural del IEEE 802, el comité eligió el protocolo paso de testigo en anillo como punto de inicio para el protocolo FDDI MAC². Ya que FDDI es además compatible con la interface común 802.2

2.4.1. ETHERNET: RED LOCAL CSMA/CD

Las especificaciones del protocolo CSMA/CD o Ethernet (estándar IEEE 802.3) se ajustan al esquema de capas. El protocolo CSMA/CD tiene las siguientes características:

1. Detección de portadoras; sólo se transmite cuando no hay energía en el medio.
2. Detección de colisión durante la transmisión; al detectar una colisión, la transmisión se aborta y reprograma.

Los mecanismos que se implantan para el acceso a Ethernet (o en el estándar CSMA/CD) es el siguiente:

La capa física es responsable de detectar el tráfico en el medio (cable) y de indicarlo a través de una señal de detección de portadora al control de acceso al medio. Esta capa también compara la señal en el medio con la señal que se genera en el caso de una transmisión, y emite a su vez una señal de detección de colisión si hay indicios de contienda en el canal. Estas funciones son realizadas por la subcapa de acceso al canal de la capa física que se muestra en la figura 2.10. El bloque de acceso de transmisión al canal generara la señal de detección de colisión; para generar la señal de detección de portadora se usan los bloques de acceso de recepción al canal y de acceso de transmisión al canal. Ambas señales son interceptadas por la capa de control de acceso al medio de enlace de datos.



Figura 2.10 Funciones de la capa física de Ethernet

² Ross E. Floyd, "An overview of FDDI: the fiber distributed data interface", IEEE journal on select areas in communications, USA, vol 7, No 7 septiembre de 1989

Además de la detección de las dos señales, colisión y portadora, el bloque de acceso al canal transmite bits hacia el medio y a su vez los recibe de éste. El bloque de transmisión de datos codificados de la capa física de datos (figura 2.10) codifica mediante el código Manchester los bits de datos.

Los datos se transmiten por la red local en una trama que incluye campos de dirección y de detección de errores, además de los datos que transmiten las capas superiores. El formato de la trama en Ethernet está representado en la figura 2.11. Las tramas tiene una longitud máxima de 1518 bytes. Los 12 primeros bytes contiene las direcciones de destino y fuente. El tipo de campo de 2 bytes está reservado para usarse en capas superiores, y no es interpretado por el protocolo Ethernet. La secuencia de verificación de trama de 4 bytes al final de la trama se usa para detectar errores.

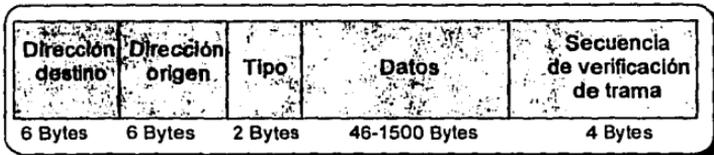


Figura 2.11 Formato de trama en Ethernet

En la figura 2.12 se muestra cómo se organiza el control de acceso al medio para realizar estas funciones. Los bloques de organización del enlace se valen de la señal de detección de portadora para realizar la prevención de la colisión, y de la señal de detección de la colisión para efectuarla prevención de la contienda. Las funciones con respecto a las tramas, direcciones y detección de errores son realizadas por los bloques de encapsulamiento de datos.

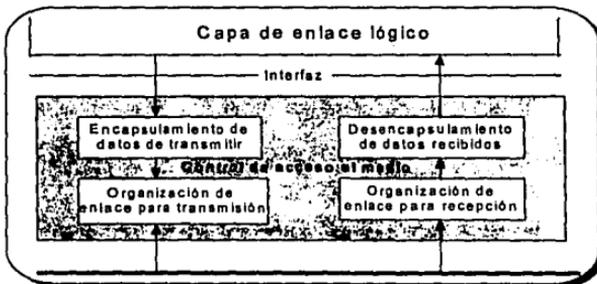


Figura 2.12 Control de acceso al medio Ethernet

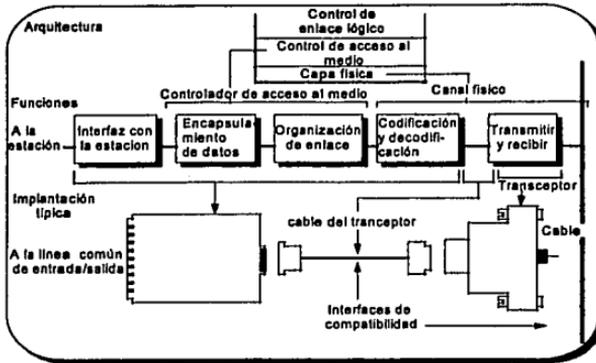


Figura 2.13 Arquitectura Ethernet e implementación típica

En la figura 2.13 aparece una implementación típica de Ethernet, que se apega a la arquitectura recién descrita.

2.4.2. TOKEN RING (PASO DE TESTIGO EN ANILLO)

El paso de señal en anillo también ha sido estandarizado como un método de acceso por el comité IEEE 802, y aparece como el estándar IEEE 802.5 [IEEE 84]. Como ya se mencionó, el estándar está basado en la técnica de paso de señal en anillo. Si el anillo está libre, irá circulando por el anillo un testigo "libre", de un nodo a otro. El testigo es el que controla el uso del anillo, indicando si está ocupado o no.

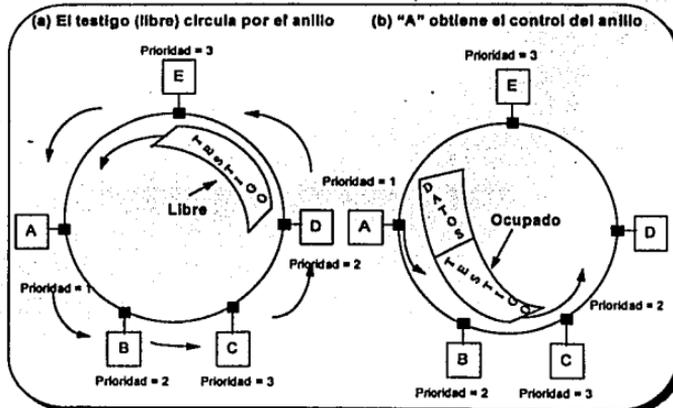


Figura 2.14 Token Ring (Red con paso de testigo en anillo)

Durante el periodo en que una estación posee el testigo adquiere el control absoluto del anillo. Una vez capturado el testigo (testigo ocupado), la estación transmisora (A en la figura 2.14) insertará datos detrás del testigo y enviará este corriente de datos por el anillo. A medida que vayan monitoriando los datos cada una de las estaciones regenerará la señal, examinará la dirección situada en la cabecera de los datos y los transferirá a la siguiente estación. Únicamente copiará sus datos si éstos deben ser entregados a la aplicación de usuario conectada a es nodo concreto. Cuando la información regrese de nuevo al nodo de partida, el testigo volverá a inicializarse (se declara libre el testigo) y se insertará a la red.

En el esquema de entrega de testigo con prioridades, cada estación posee una determinada prioridad de acceso a la red, esta condición se expresa colocando en el testigo indicadores de preferencia.

En la figura 2.15 aparece el formato de una trama transmitida por el anillo. El delimitador de principio DP y el delimitador de fin DF, de un solo byte, denotan el principio y final de la trama respectivamente. El control de acceso de 8 bits o campo CA se usa para implantar el protocolo de acceso al anillo. El bit T se pone en cero cuando se trata de un testigo libre y en uno cuando el testigo esta ocupado. Los tres bits P en el campo CA se usan para proporcionar hasta ocho niveles de prioridad en el acceso al anillo.

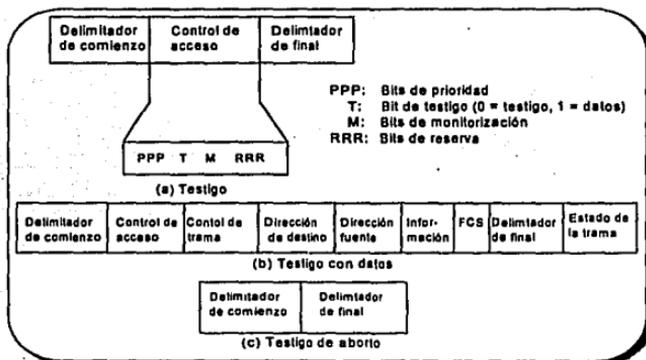


Figura 2.15 Token Ring (Formato de anillo IEEE 802.5 paso de testigo en anillo)

El mecanismo de prioridad funciona de modo que todas las estaciones con el mismo nivel de prioridad tengan las mismas oportunidades de acceder al anillo. Ello se logra haciendo que la estación que elevó el nivel de prioridad necesario para acceder al anillo (estación depositaria) vuelva a dejarlo como estaba al terminar. Los tres bits R en el campo CA se usan para reservar el uso del testigo que se transmita con la prioridad solicitada. Estos tres bits se pueden activar en el momento de repetir el testigo ocupado o libre.

Los otros campos del formato de la trama se identifican claramente. Las direcciones de destino y origen son de longitudes de 2 o 6 bytes. Se puede dirigir mensajes a estaciones individuales o grupos de estaciones. La dirección de destino con sólo unos se usa para difundir información a todos los nodos del anillo. El campo de

secuencia de verificación de trama de 4 bytes proporciona la detección de errores de bit. El protocolo de acceso para paso de señal en anillo no corrige los errores , sólo informa al control de enlace lógico.

El estándar IEEE 802.5 permite tres formatos distintos de red en anillo [IEEE 84], que son los que aparecen en la figura 2.15. El formato de testigo figura 2.15 a. El formato de transferencia fig. 2.15 b. En la fig. 2.15 c puede verse un testigo de aborto, formado únicamente por los delimitadores de inicio y fin, este tipo de testigo puede enviarse en cualquier momento para abortar una transmisión previa.

2.4.3. INTERFAZ DE DATOS DISTRIBUIDA DE FIBRA (FDDI)

La Interfaz de Datos Distribuida de Fibra (Fiber Distributed Data Interface, FDDI) es un estándar de cable de Fibra Optica desarrollada por el comité X3T9.5 del American National Standards Institute (ANSI). Trabaja a 100 mbps y utiliza una topología en anillo doble y rotación inversa. Si falla un enlace de anillo, el anillo se reconfigura de modo que se puede seguir aceptando trafico en la red hasta que se corrige el error.

FDDI es un método de Acceso de Paso de Testigo. Se pasa un mensaje testigo de estación a estación a lo largo de la red. Si una estación necesita transmitir, coge el testigo. Las estaciones actúan como repetidores, transmitiendo los paquetes que reciben a la siguiente estación. En la figura 2.16 se observa que los componentes (Terminales) están interconectados a través de un concentrador, que sirve de punto de encuentro y reconfiguración para todas las líneas de fibra óptica y para todo el flujo de datos. El canal interno enlaza sólo determinados dispositivos. Estos dispositivos, que tienen conectados los anillos interno y externo, tienen la clasificación A (estaciones críticas que necesiten apoyo adicional y canales de mayor velocidad). Los dispositivos tipo B (estaciones de menor importancia) sólo están unido a un anillo.

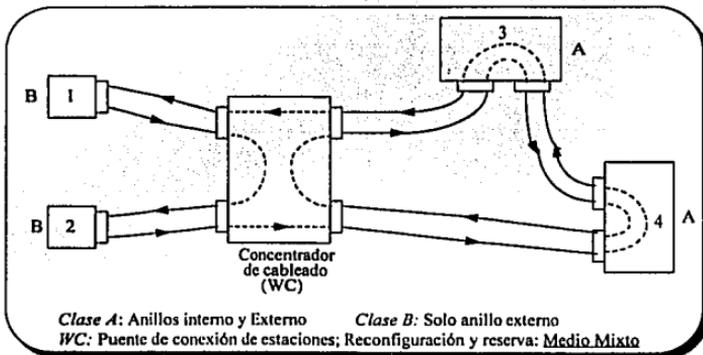


Figura 2.16 Interfaz distribuida de datos basada en fibra óptica

El concentrador permite conectar estaciones y reconfigurar el sistema. también se encarga de aislar los nodos problemáticos mediante el punto de concentración. El concentrador puede incluir un interfaz en el que el

usuario instalará fibra óptica para una parte de la red, y cable coaxial o par trenzado (telefónico) para otra región de la misma.

Los conectores de los terminales y del concentrador son diodos láser. El número máximo de estaciones en una red es de 500. Hay cable de Fibra Óptica monomodal y multimodal. El cable monomodal deja pasar una frecuencia luminosa, y el multimodal deja pasar varias frecuencias. Las redes FDDI pueden abarcar áreas de aproximadamente 200 Km. La distancia máxima entre estaciones es de 2 Km. con cable multimodal y 60 Km. cuando se usa cable monomodal.

2.4.4 PROTOCOLO X.25

El protocolo X.25 cubre la conexión de terminales de datos, computadores y otros sistemas de usuario o dispositivos para redes de conmutación de paquetes. Los sistemas de usuario se llaman equipo terminal de datos (ETD), que pueden ser computadoras, terminales o impresoras. Su conexión a la red se hace por medio de un equipo de red llamado Equipo de terminación de circuitos de datos (ETCD), que pueden ser Modems o convertidor de señales. Normalmente, un ETD desea establecer comunicación con otro ETD y utiliza la red con ese propósito. A su vez, ese ETD se conectará a un ETCD que controla su acceso a la red, y esta será responsable de administrar las comunicaciones entre los ETCD. El protocolo X.25 regula el flujo de datos entre ETD y ETCD solo en cada extremo de la red. En la figura 2.17 se muestra este concepto en forma gráfica.

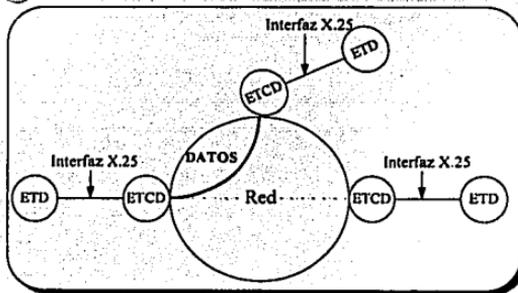


Figura 2.17. Concepto de X.25

Así, X.25 es solo una especificación de interfaz, gobierna las iteraciones entre un ETD y el ETCD al que esta conectado. Los detalles de la comunicación entre ETCD, se realizan usando la red que los conecta, dejando al operador la ésta. Tales detalles quedan ocultos a los ETD. El protocolo X.25 esta organizado como una arquitectura de tres niveles (de red) del modelo OSI. En la figura 2.18 ilustra los tres niveles de X.25, en tanto que en la figura 2.19 se muestra la relación nivel por nivel, con la arquitectura OSI.

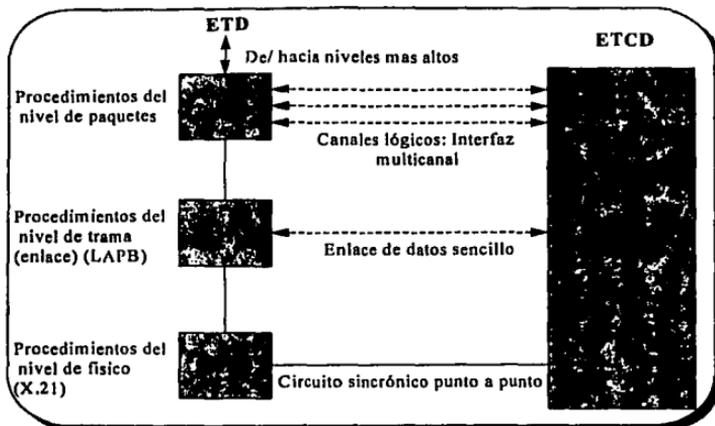


Figura 2.18 Capas X.25

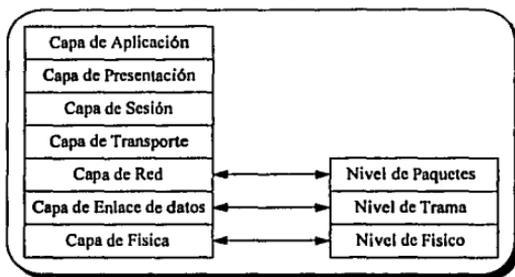


Figura 2.19 Relación entre la Arquitectura OSI y X.25

En la figura 2.18 se muestra los niveles de X.25. Como en el modelo OSI, el nivel físico más bajo asegura la existencia de una conexión física válida entre el ETD y el ETCD, es recomendado X.21 para este propósito. El protocolo en el nivel de enlaces de datos es un subconjunto de HDLC (High Level Data Link/ Nivel alto del control de enlace de datos), etiquetado como LAPB (Balanced link access procedures/ procedimiento balanceados de accesos). Las unidades de datos o tramas que atraviesan el enlace en cualquier dirección, de ETD a ETCD tiene el procedimiento mostrado en la figura 2.20. El principio y el final de la trama se denomina con un campo de sincronía de 8 bits especial F, 01111110. A es un campo de dirección; C, un campo de control, FCS, un campo que se utiliza para la detección de errores y el campo I, o de información, lleva los datos (el paquete) transferidos del nivel de red.

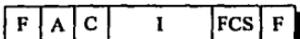


Figura 2.20 Formato del Trama HDLC

En el tercer nivel de red, donde el protocolo de X.25 se distingue como una arquitectura interfaz. A esta capa se llama nivel de paquetes en la terminología de X.25, enfocándose a conexiones de circuito virtuales.

Cada Interfaz asigna su propio ETD-ETCD asignando su propio conjunto de números de canales lógicos. Así, un Circuito Virtual completo, de extremo a extremo entre dos ETD que se comunican entre sí, pueden usar diferentes números de canales lógicos en las dos interfaces en cada extremo del circuito virtual. (En la figura 2.17 se delimitan dos de estos Circuitos Virtuales).

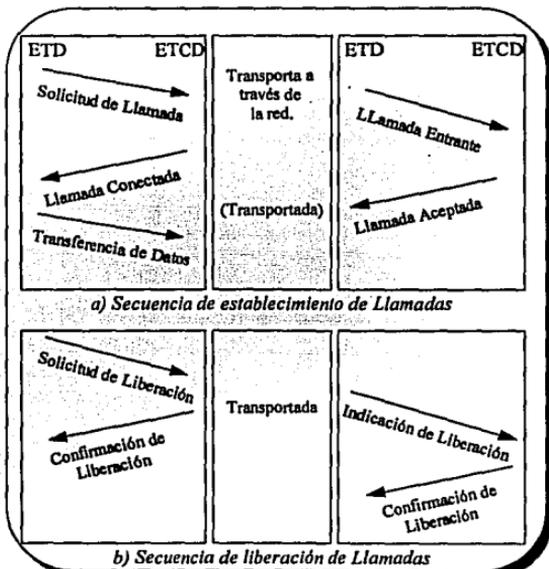


Figura 2.21 Secuencia de establecimiento y liberación de llamadas

Como en la Arquitectura del modelo OSI, el protocolo X.25 requiere 3 fases de comunicación para la operación de CV (Circuitos Virtuales). Estos son en orden: la fase de establecimiento de la llamada o simplemente fase de establecimiento; la fase de transferencia de datos y la fase de desconexión o liberación de la llamada (se pueden utilizar CV permanentes, en cuyo caso no se necesitan establecer la llamada). En la figura 2.21 se encuentra diagramada la fase de establecimiento. Las flechas indican los diversos paquetes de control

(unidades de datos del protocolo de nivel de red) que se transmiten entre ETD y el ETCD al que este se halla conectado. Estas son todas las unidades de datos de igual nivel que se transmiten entre protocolos de nivel de paquetes en cada uno de los dos sistemas (ETD y ETCD). Los paquetes de la figura 2.21 son unidades de datos en el mismo nivel de red que se transportan entre dos sistemas distintos, y por ello se pueden distinguir desde el exterior.

El paquete de solicitud de llamada de X.25 de la figura 2.21 notifica al ETCD al que se dirige que el ETD que lo emite desea establecer comunicación con otro ETD conectado a la red. El paquete lleva un número de canal lógico para que lo usen en las conexiones lógicas ETD-ETCD, así como en las direcciones de los ETD emisor y receptor, los parámetros que especifican las características de la llamada, más 16 bytes de datos como máximo. Esta información se transmite a través de la red usando los procedimientos propios de ella, y alcanza al ETCD. Luego, este sistema transmite el mismo tipo de paquete que lleva la misma información, excepto quizás por el número de canal lógico diferente, al ETD destino (receptor). Como este paquete lo emite un ETCD se cambia su nombre a paquete de llamada entrante. El ETD contesta con un paquete de llamada aceptada, el cual, al avanzar a través de la red, aparece como un paquete de llamada conectada, emitido por el ETD emisor como respuesta a las peticiones originales de llamada de los ETD emisores. Ahora puede empezar la transferencia de datos usando paquetes de datos (UDP de datos de nivel de red). El procedimiento de liberación de la llamada es similar, como se muestra en la figura 2.21; la diferencia es que el ETCD receptora acepta inmediatamente al paquete de solicitud de liberación, sin esperar una confirmación de otro ETD.

Igual que el modelo OSI, el protocolo X.25 tiene control de errores y de flujo al nivel de paquetes (red), para la transmisión expedida de datos y para otras operaciones similares a las incorporadas en los protocolos de los niveles OSI.

2.5 PROTOCOLOS DE NIVELES SUPERIORES

En esta sección se presenta una revisión de los principales protocolos usados en comunicaciones cliente/servidor. Estos protocolos se ubican en el modelo OSI a partir de la capa de red. En la figura 2.22 se muestra la ubicación en el modelo OSI de los principales protocolos usados en las comunicaciones cliente/servidor dependiendo del sistema operativo de red.

Por cuestiones de alcance del presente trabajo se revisaran únicamente los protocolos TCP/IP y NetBIOS que son lo suficientemente representativos de los protocolos de la figura 2.22. También es importante destacar que en la actualidad los sistemas operativos de red ofrecen (ver capítulo 4) facilidades para el intercambio de datos entre distintos protocolos.

OSI	NETWARE	UNIX	APPLE	LAN MANAGER
APLICACION	NETWORK CORE PROTOCOL		APPLE SHARE	BLOQUES DE MENSAJES DEL SERVIDOR
PRESENTACION			APPLE TALK FILING PROTOCOL (AFP)	
SESION	NAME PIPEB	NETBIOS	ASPADSP ZIP PAP	NETBIOS
TRANSPORTE	SPX		NDP RTMP	NAMED PIPEB
RED	IPX		NETBEUI	
ENLACE DE DATOS	CONTROLADORES DE RED		CONTROLADORES DE RED	
	ODI	NDIS	LOCAL TALK ETHER TALK TOKEN TALK	
FISICO	FISICO		FISICO	

Figura 2.22 Protocolos de Sistemas Operativos

2.7.1. TCP/IP

Los ambientes que utiliza TCP/IP se basa en que cada elemento de la red tenga su dirección IP. Para IP tenemos un número formado de 4 bytes.

El propósito de ambos casos es el mismo: Identificar de forma única a cada elemento del conjunto. En la terminología normal de TCP/IP a los nodos que son computadoras, se les denomina genéricamente Host (No confundir con la aceptación normal bajo la cual la palabra Host significa un equipo mayor, típicamente un mainframe). Y también en la misma terminología se le denomina gateway a un dispositivo que tiene realmente funciones de ruteador o enruteador.

Como el uso de direcciones es muy frecuente sería un poco difícil escribir números de 32 bits e incluso tener que expresarlos como :

2AB1CC02,

es decir en forma hexadecimal. Afortunadamente se ha escogido una forma decimal, en la que el valor de cada byte se separa por un punto:

129.3.5.88

El numero anterior representa la dirección IP de un determinado red. Observe la siguiente ilustración ejemplificando las direcciones IP:

En la figura 2.23 se puede apreciar que los nodos de la misma red, tiene el mismo prefijo, sin embargo las particulares de la clase de direcciones IP son diferentes.

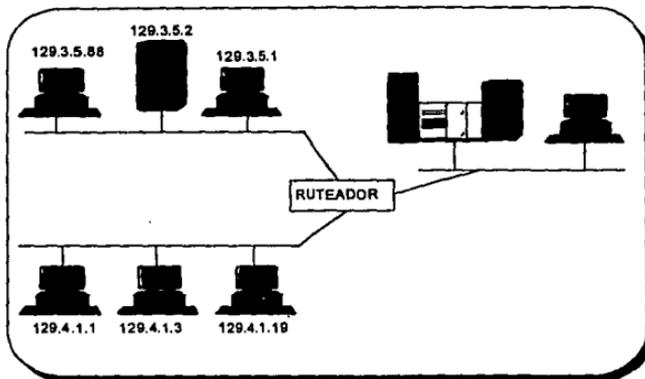


Figura 2.23 Ejemplo de uso de direcciones IP

Las direcciones IP sirven para:

1. Identificar la manera única de cada nodo de una red o de un grupo de redes(Inter-red).
2. Identificar también a miembros de la red (debido a que tiene el mismo prefijo).
3. Poder dirigir o direccionar información entre el nodo y otro, a un cuando ambos estén en distintas redes.
4. Para dirigir o direccionar información a todos los miembros de una red o un grupo de redes.

Al enviar un paquete de información a través de IP y a otra red, se necesita primero enviarlo al Ruteador para que este a su vez lo envíe a su destino.

El protocolo IP no contempla ningún tipo de mecanismo para saber si un paquete llegó o no. Así mismo no tiene un mecanismo para regular el flujo de paquetes. Es decir, que prevenga el hecho de que un nodo muy veloz pueda saturar de paquetes a otro mucho menos rápido.

Ambos mecanismos de los que carece IP están contemplados en TCP, y son llamados mas formalmente:

1. Secuenciamiento y reconocimiento de paquetes.
2. Control de flujo de información.

El secuenciamiento es parte de la primera tarea significa que cada mensaje que un nodo desea enviar, lo partirá en paquetes. A cada paquete le asignará un número (por ejemplo: paquete 1, paquete 2, 3, 4, etc.).

El reconocimiento es un complemento de lo anterior. Significa que cuando un nodo ha recibido varios paquetes, le debe informar al que los esta enviando que efectivamente los ha recibido. Este acuse de recibo lo hace TCP diciéndole que tipo de paquetes es el siguiente que le será enviado.

La información del número de secuencia del paquete, del reconocimiento, del control de flujo está contenida en el encabezado (Header) de TCP.

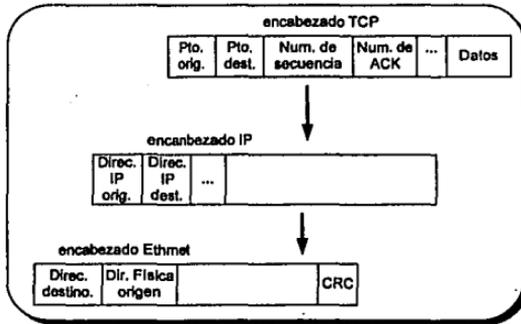


Figura 2.24 Encabezado simplificado de TCP e IP

Las tareas que hace TCP es secuenciar paquetes, enviar los avisos de reconocimiento (ACKs) y controlar el flujo son posibles gracias a que antes que se envíe información entre dos nodos, TCP establece una conexión. Este proceso también es llamado saludo (Handshake) y puede ser simplificado en tres pasos:

1. El nodo A avisa al nodo B el envío de Información
2. El nodo B responde al nodo A que acepta la información
3. El nodo A al nodo B transmite.

Y cuando se ha terminado de enviar paquetes, se hace el proceso contrario para terminar la conexión, igualmente en 3 pasos. Por lo anterior, TCP se denomina un protocolo orientado a conexión mientras IP es un protocolo no orientado a conexión.

LOS SERVICIOS ESTANDARES TELNET Y FTP

Los tareas fundamentales a nivel usuario de TCP/IP son:

1. La emulación de terminales, para poder entrar a una diversidad de equipos, para esta tarea es utilizado Telnet.
2. La transferencia de archivos entre una terminal y otra. Para esta tarea es utilizado FTP (File Transfer Protocol).

Telnet y FTP son servicios, utilerías y protocolos de TCP/IP. Son programas por los usuarios, se pueden catalogar como utilerías y como tales nos dan ciertos servicios, pero también define la forma de intercambiar información entre el equipo que solicita el servicio y el que lo brinda, entonces pueden ser llamados protocolos:

Como protocolos Telnet, FTP y algunos otros (SMTP) siempre se usan en TCP/IP y colocándose en los niveles de 5, 6 y 7.

SERVICIOS DE RED

En algunas ocasiones estos protocolos no son los mas adecuados por lo tanto existen otros protocolos para acceder a los servicios de otras máquinas y tal es el caso de NFS, Lan Manager o Netware.

NFS (Network File System) es un servicio orientado a que usuario de equipo con Sistemas Operativos distintos puedan ver de una manera mas simple archivos y programas almacenados en cualquiera de los servidores de una inter-red. De manera que logra una mayor transparencia que con FTP. La mayoría de los equipos que soportan TCP/IP, tienen implantados NFS.

NFS utiliza un protocolo denominado RPC (Remote Procedure Call. Explicado en el capitulo 3) el cual a su vez se monta sobre UDP (User Datagram Protocol). UDP actúa en el mismo nivel que TCP, pero no esta orientado a conexión.

Aunque NFS brinda ya servicios de archivos distribuidos, se queda corto en algunas tareas como la compartición de los mismos (Acceso concurrente), servicios de seguridad y contabilidad de usuarios.

En este caso se requiere de un Sistema Operativo de red instalado en varios equipos de la red, y que pueda dar todos estos servicios. En cualquier caso, no importa si el equipo es un PC-386 o un mainframe, decimos que es un servidor de red.

De acuerdo al tipo de equipo y sistema operativo, tenemos dos casos:

1. El sistema operativo de la red corre como una tarea del sistema operativo nativo del equipo.
2. El sistema operativo de red corre en forma nativa en el equipo (Sin algún otros sistema operativo).

En el primer caso tenemos, por ejemplo, a Lan Manager como sistema operativo de red, y a Unix, OS/2 y Windows NT como sistema operativo nativo. También en este caso se encuentra Unix con Portable-Netware.

La ventaja de TCP/IP de LanManager vs Netware es que el soporte real es éstos protocolos es mucho mejor. En el caso de Lan-Manager (para Unix o para OS/2) el juego de protocolos de TCP/IP queda montado en cada PC de la red, además del servidor, y resulta mas transparente su uso, en comparación con Novell-Netware que se requiere de un traducción muy especial. Sin embargo si se requiere de servicios de NFS en los servidores Netware se encuentra mas potente.

SERVICIOS DE BASES DE DATOS TCP/IP

Dentro de las aplicaciones Cliente/Servidor, una de las que mayor auge ha tenido ha sido la de Base de Datos: Teniendo por un lado el equipo corriendo el manejador de base de Datos, y por otro, muchas PC 's conectándose a él, a través de diversas herramientas e interactuando con la información.

Las aplicaciones que corren en las PCs se denominan "clientes", y el equipo que tienen las bases de datos se le denomina "Servidor". Como se desea realizar una conexión entre clientes y servidor(es) no importando si estos están en la misma red o en redes distantes, la solución más sencilla es que ambos: clientes y servidores tengan una comunicación usando TCP/IP

2.7.2. NetBIOS

NetBIOS define una interface en la capa de sesión para NetBEUI's que se encuentra funcionalmente la capa de transporte y red. NetBIOS se implementa dentro de varios ambientes de redes incluyendo Microsoft® MS-Net, LAN Manager, IBM's PC Network y Lan Server. En algunos ambientes de red, la interface de sesión NetBIOS es usada por algunos protocolos diferentes en las capas de transporte y red. Por ejemplo en LAN Manager, La interface NetBIOS puede ser usada para acceder al mecanismo de transporte NetBEUI, TCP/IP o XNS. En Netware, la interface NetBIOS es usada para acceder ya sea SPX/IPX o TCP/IP. Esto es mostrado en la figura 2.25.

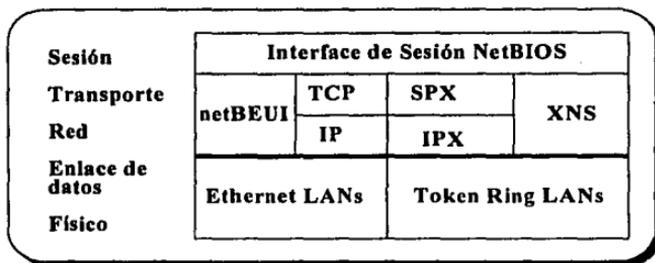


Figura 2.25 Interfaces comunes usadas abajo de la capa de sesión con NetBios

NetBIOS ofrece servicios de IPC que pueden ser usados directamente para implementar cliente/servidor, procesamiento filtro o comunicación punto-punto. NetBIOS ofrece servicios de comunicación orientados a conexión y no orientados a conexión. Además proporciona un servicio de localización de nombres para facilitar el encuentro de un proceso específico.

COMANDOS

Todos los servicios de NetBIOS son ofrecidos como comandos formateados dentro de una estructura común de datos, llamada Bloque de Control de Red (Network Control Block NCB). Los formatos de la aplicación y los campos de el NCB pasan a NetBIOS a través de una sola llamada al sistema (ej. INT 2A₁₆ en MS-DOS o NetBIOSsubmit en OS/2). El NBC tiene una longitud de 64 bytes. NetBIOS usa el NBC para construir frames NetBEUI, los cuales son encapsulados en paquetes de red (ej. Ethernet).

En un sistema multitareas tal como OS/2 o UNIX. Los comandos NetBIOS pueden ser con bloqueo (WAIT modo) o sin bloqueo (NO_WAIT modo). Diferentes códigos de comando NBC corresponden a los modos de bloqueo y sin bloqueo para un mismo comando. Por ejemplo el comando SEND con bloqueo tiene un código de comando de 14₁₆ y el comando SEND sin bloqueo tiene un código 94₁₆. El código retornado en el NCB indica el éxito o falla para comandos con bloqueo y un estatus de "formado" para comandos sin bloqueo.

La terminación del campo de estatus de el NBC debe ser examinado para determinar cuando un comando sin bloqueo en realidad se ha completado.

ORIENTADO A CONEXION.

En una sesión orientada a conexión, un número de sesión, asignado durante el establecimiento de la conexión es usado por los subsecuentes comandos NBC dentro de esta sesión. Cada proceso puede tener hasta 254 sesiones concurrentes, usando el número de sesión para distinguir entre ellos. Para aceptar una sesión orientada a conexión, un servidor remite un comando LISTEN indicando que aceptara cada uno de todas las llamadas o una llamada específica (por nombre). Para iniciar una conexión, un cliente invoca el comando CALL indicando su nombre y el del servidor. Una vez que NetBIOS relaciona el CALL con el LISTEN, una sesión es establecida y cada extremo puede enviar y recibir datos usando SEND y RECEIVE enviando hasta 64KB de datos en cada SEND. Además se pueden encadenar dos SEND si son demasiado largos los mensaje que deben ser enviados. Como la mayoría de servicios orientados a conexión, NetBIOS asegurará que cada mensaje enviado sea recibido en orden. Para acelerar la transmisión, NetBIOS permite la opción de no esperar el ACKS después del envío de mensajes. Esto es usado frecuentemente cuando varios SENDs serán transmitido a la vez (ej. Copiar un archivo)

NO ORIENTADO A CONEXION

Cada proceso puede agregar hasta 254 nombres lógicos a la tabla de nombres de NetBIOS (indexado por el número del nombre en el NCB). Estos nombres pueden ser usados como nombres globales en datagramas. NetBIOS soporta nombres únicos y nombres agrupados. Cuando un nombre es agregado a la tabla, NetBIOS transmitirá el nombre para advertir su identidad y localización. Además asegurara que un nombre único no sea usado por otros y checa que nadie este usando un nombre de grupo como nombre único.

En un modo sin conexión los datagramas pueden ser enviados a la locación de un nombre (usando un nombre único en la tabla), multitransmisión a grupos (usando nombres de grupo) o transmitiendo a todos los nodos en la red. Los datagramas no están reconocidos y pueden perderse, recibirse fuera de orden o aun recibirse doble vez (aunque esto es raro). La terminación exitosa del comando SEND DATAGRAM simplemente significa que el paquete fue exitosamente enviado a buscar su extremo en la red. Un receptor no puede recibir datagramas sólo desde un específico emisor de nombres. Un receptor debe aceptar datagramas desde cualquier terminal que envía paquetes a él. Con especificar un nombre FF₁₆, un receptor puede recibir copias de todos los datagramas enviados, aun aquellos direccionados a otros NetBIOS limita el tamaño máximo de datagramas a 512 bytes.

EJEMPLO

La figura 2.26 muestra una típica sesión cliente/servidor usando NetBIOS. El cliente emite un ADDNAME para registrarse el mismo, seguido por un CALL para solicitar una sesión con su servidor. Si el servidor acepta al cliente, el cliente obtiene un número de sesión para usar con subsecuentes operaciones SEND

y RECEIVE. En forma similar, el servidor emite un ADDNAME cuando inicia, seguido por un comando LISTEN. Si el servidor acepta al cliente, él obtiene el número de sesión de los CALL's y emite un RECEIVE para aceptar mensajes. Cuando un mensaje llega, el servidor puede responder usando un SEND y el número de sesión apropiado. Un comando HANGUP deshace la sesión en ambos lados.

Si el mismo ejemplo estuviera usando datagramas, las operaciones CALL, LLISTEN y HANGUP tendrían que ser removidas de la figura 2.26, ya que no existe el concepto de "sesión" cuando se usan datagramas, también deberá removerse el número de sesión. Todo lo demás permanecerá en la misma forma.

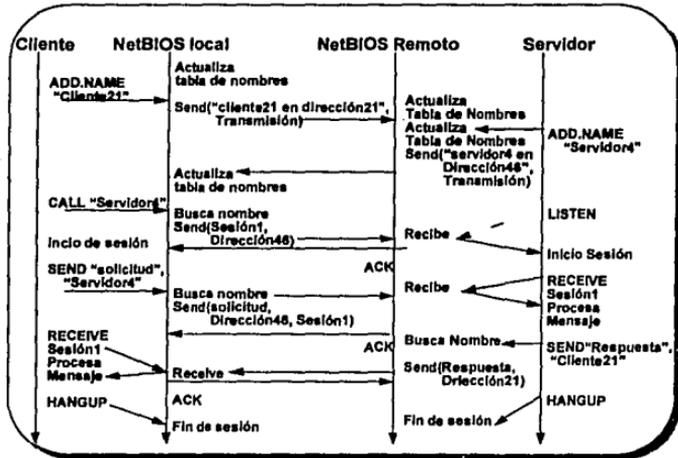


Figura 2.26 Ejemplo de una sesión Cliente/Servidor NetBios

2.6 PUENTES Y RUTEADORES

La motivación principal por la que surgen Puentes y Ruteadores, es la de intercomunicar redes que antes estaban separadas, o bien para poder separar en redes pequeñas, una red mayor.

Si hablamos del modelo OSI, un puente trabaja en el nivel 2 y un ruteador trabaja en el nivel 3.

2.6.1. PUENTES

FUNCIONAMIENTO INTERNO

Fundamentalmente un puente es un dispositivo que "ve" tramas y decide si los debe pasar de un lado al otro. A la labor de "ver" tramas se le llama *capacidad de filtraje*, mientras que a la de pasar los frames de una red a otra se le denomina *capacidad de reenvío* (forwarding).

Para que el puente pueda hacer sus labores de reenvío de tramas, debe conocer que direcciones están en cada uno de sus puertos.

En la figura 2.27, los pasos que tiene que seguir el puente son los siguientes:

- Se supone que ya tiene construida la tabla de direcciones (de algún modo).
- Cuando llega una trama por el puerto A, se verifica su dirección destino.
- Si la dirección destino se encuentra en la lista de aquellas del puerto B, la trama se envía para el otro puerto. De lo contrario, la dirección es de algún nodo de la misma red, y no se hace nada con la trama (este caso no se ilustra en la figura 2.27).

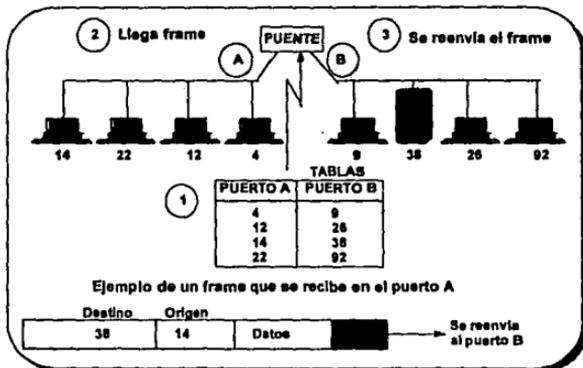


Figura 2.27 Forma de trabajo de un puente (como la trama que se recibe en A tiene una dirección destino que está en B, la trama se reenvía a B)

LAS TABLAS DE DIRECCIONES.

El puente tiene sus tablas de direcciones y éstas fundamentalmente tiene tres maneras de actualizarse:

1. De forma estática.- Es decir, introducir las manualmente vía una micro o terminal, conectada al puente.
2. De forma dinámica.- Bajo este esquema, si el puente "ve" una trama cuya dirección origen no tiene en sus tablas, actualiza la tabla del puerto en el que vio dicha trama. A esta labor se le denomina *Aprendizaje de direcciones*, y algunos fabricantes llaman inteligentes a los puentes que la realizan.
3. Una mezcla de ambas.

2.4.2. RUTEADORES

FUNCIONAMIENTO INTERNO.

A diferencia de los puentes, los ruteadores trabajan en función de las direcciones del protocolo de nivel 3 (nivel de red).

La figura 2.28 muestra un ejemplo de ruteadores en ambiente TCP/IP.

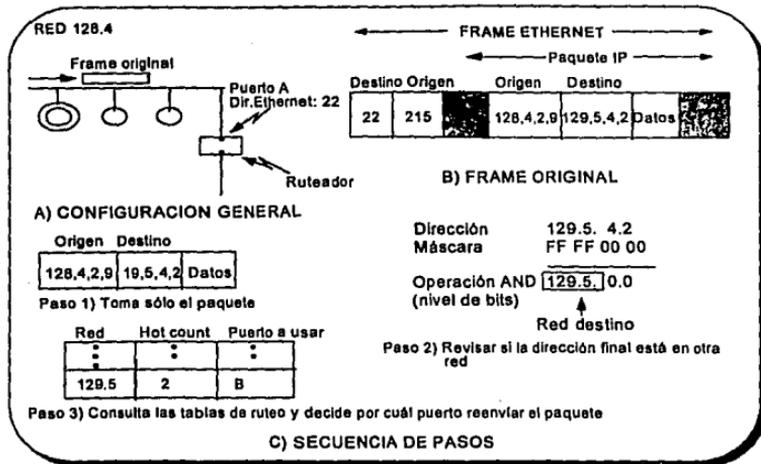


Figura 2.28 Pasos fundamentales de un ruteador en el reenvío de paquetes

Obsérvese que lo único que se necesita saber en sus tablas qué números de redes existen en toda la inter-red, no requerimos llevar la relación de todas las direcciones físicas.

El ruteador hace lo equivalente al puente, pero a nivel 3:

1. Toma el paquete que estaba dentro de una trama, y desecha la trama.
2. Revisa si la dirección final está en otra red. Por ejemplo, el ruteador recibe un paquete que viene de la red 128.4 y el paquete va para la 129.5
3. Consulta entonces sus tablas de ruteo (previamente construidas), y sabe por cual de sus puertos enviar el paquete. Puede ser que el paquete sea enviado a su red destino, o requiera de ir a otro ruteador en donde sucederán estos mismos pasos.

Los pasos anteriores están representados en la figura 2.28.

La realización de la segunda tarea (verificar la dirección para conocer si es de otra red) nos obliga a tener un *protocolo ruteable*. Es decir, un protocolo que maneje direcciones de al menos dos coordenadas: Número de Red, Número de Nodo.

Los protocolos ruteables más típicos son: IP (de la familia TCP/IP), IPX (de Novell) y XNS. El *protocolo-no ruteable* mas popular es NetBios, pues sólo existe un nodo si está en la misma red.

LAS TABLAS Y LOS ALGORITMOS DE RUTEO.

Para la labor de un ruteador, necesita consulta sus tablas *previamente construidas*. Dichas tablas se construyen con base en un algoritmo de ruteo, que define dos cosa fundamentalmente: El tipo de información

que se van a intercambiar los ruteadores, y bajo qué circunstancias se la envían (por tiempo definido y/o por evento).

Los parámetros y criterios que se considerarán para elegir la mejor ruta, a dichos criterios se les denomina métricas. En general entre más métricas se tomen en cuenta, mejor es el algoritmo.

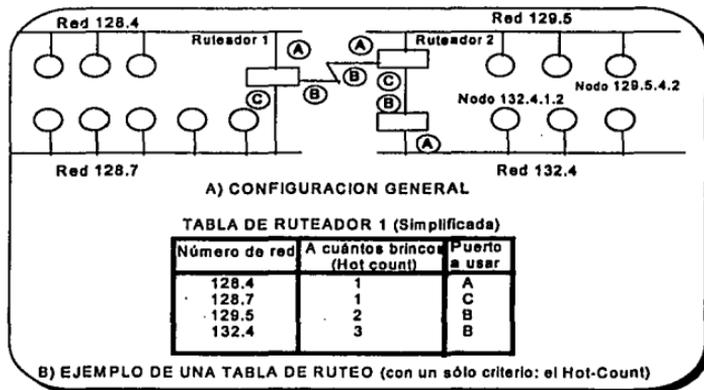


Figura 2.29 Ruteadores en un ambiente TCP/IP

Los protocolos de ruteo más comunes son RIP (Routing Information Protocol) y OSPF (Open Shortes Path First). Sin embargo RIP es limitado para redes grandes con muchos ruteadores por varias razones, entre las principales:

1. **Sólo considera una métrica:** La distancia "en brincos" que hay de una red a otra, (Hot Count). Es decir, cuantos ruteadores hay que atravesar. Y la Lógica es: a menos ruteadores, mejor ruta. No importa si la ruta que tomó es un canal a 9600 bps, en vez de ir por otra que usa un ruteador más, pero va por canales de 2 Mbps.
2. **No optimiza el envío de tablas de ruteo:** Cada determinados segundos, todos los ruteadores se envían sus tablas de ruteo completas.
3. **El tiempo de "convergencia" es alto.** Es decir, cuando existe algún cambio repentino en la topología de la inter-red (por ejemplo se cayó un enlace), el tiempo que tardan todos los ruteadores en cambiar sus tablas y reflejar dicho cambio, es mucho mayor que otros protocolos de ruteo.
4. **No puede balancear cargar:** Si una ruta está muy congestionada, el protocolo, no contempla algún tipo de enrutamiento alterno temporal.

OSPF en cambio, supera todas estas limitaciones:

1. Considera muchas más métricas.
2. El envío de tablas se basa en una forma de jerarquización de ruteadores.

3. Dicho envío está optimizada de acuerdo a los cambios de topología.

4. Permite un balanceo de cargas.

O sea, todo lo que RIP ni hacía. Sin embargo para redes que tengan unos cuantos ruteadores, RIP funciona bien.

Dentro de los protocolos de ruteo comerciales existe también IGRP, pero como es propietario de Cisco sólo ellos lo implementan.

Todos los protocolos mencionados: RIP, OSPF y IGRP son usados dentro de una inter-red. A diferencia de los protocolos *externas*, que son usados para enlazar varias inter-redes. Tal como es el caso de EGP (External Gateway Protocol)

CAPITULO III

LA ARQUITECTURA

CLIENTE/SERVIDOR

CAPITULO 3 LA ARQUITECTURA CLIENTE/SERVIDOR

Sistemas de Procesos Distribuidos

Introducción

La tecnología cliente/servidor suele ser confusa por ser un concepto relativamente nuevo. En este capítulo se tratan los fundamentos de la tecnología cliente/servidor y sus características principales. Se trata la arquitectura cliente/servidor desde la perspectiva teórica del procesamiento distribuido. Se comienza el capítulo definiendo el procesamiento distribuido, proporcionando una taxonomía del procesamiento distribuido para ubicar dentro de este contexto el paradigma llamado cliente/servidor.

Se da una descripción de la parte cliente y la parte servidor. Además se profundiza en los aspectos esenciales de comunicación entre el cliente y servidor.

La clave para entender cliente/servidor está en realizar que es un concepto lógico, la parte del cliente y el servidor pueden, o no existir en distintas máquinas físicas. La tecnología cliente/servidor es un paradigma o modelo para la interacción de procesos de software ejecutándose concurrentemente.

3.1 PROCESOS DISTRIBUIDOS

3.1.1. DEFINICION DE PROCESOS DISTRIBUIDOS

Los procesos distribuidos son procesos concurrentes que se comunican usando mecanismos de mensajes transitorios encontrando facilidades dentro de los interprocesos de comunicación (IPC, interprocess communication). Los procesos concurrentes como su nombre lo indica se ejecutan en un sistema al mismo tiempo. Se da una descripción detallada de proceso en el capítulo 4.

La mayoría de sistemas operativos concurrente proporcionan interprocesos de comunicación (IPC) que son facilidades para que los procesos puedan comunicarse unos con otros. Un tipo especial de IPC es la llamada a procedimiento remoto (RPC).

3.1.2. MODELOS DE PROCESOS DISTRIBUIDOS

Hay cuatro tipos básicos de procesos distribuidos: *filtros*, *clientes*, *servidores* y *puntos*. En la figura 3.1 se muestran estos tipos de procesos. Un procesador filtro ejecuta una operación predefinida sobre una cadena de datos, pasando los datos a otro procesador. Los procesadores punto son idénticos uno con otro e interactúan en forma cooperativa para hacer un trabajo útil.

Un error común es creer que el procesamiento "cliente/servidor" y "punto a punto" son sinónimos. Esta idea es verdadera en el sentido de comunicaciones, pero es erróneo en términos de procesamiento distribuido. Un ambiente de comunicaciones "punto a punto" se refiere a dos entidades comunicándose en condiciones de

igualdad. Sin embargo el termino "punto" en el sentido de procesamiento distribuido realmente significa clone - una extensión de los recursos de la máquina. En contraste los procesadores cliente y servidor no son clones uno respecto al otro; sino, independientes (tareas cooperando)

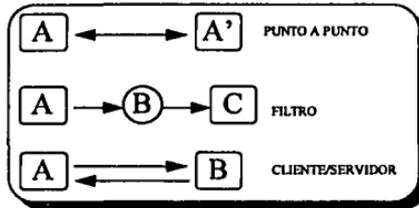


Figura 3.1 Tipos de procesamiento distribuido

El procesador cliente envía solicitudes al procesador servidor, el cual responde con los resultados a esta solicitud. Como el nombre indica el procesador servidor suministra servicios a sus clientes. La interacción entre el cliente y el servidor es cooperativa. Utiliza el intercambio transaccional en donde al cliente es proactivo y el servidor es reactivo.

En un verdadero ambiente cliente/servidor, es indiferente si los procesos del cliente y el servidor se ejecutan o no en la misma máquina. Sin embargo, estos son más interesantes cuando los procesos del cliente y servidor se ejecutan en máquinas diferentes conectadas vía red. Aunque la red puede ser de área local o de área amplia, en las redes de área local (LANs) se encuentran las más comunes implementaciones cliente/servidor.

3.1.3. ESTRUCTURA DE SISTEMAS DISTRIBUIDOS

De acuerdo a lo expresado anteriormente se usará el termino *sistema distribuido* al referirse a aplicaciones corriendo en dos o más sistemas de computadoras trabajando juntos para completar un trabajo común. El hardware de los sistemas pueden ser de diferentes arquitecturas y capacidades. Los procesos distribuidos frecuentemente unen las aplicaciones, tradicionales sistemas de información centralizadas, con el ambiente de estaciones de trabajo orientadas a los usuarios.

En general se puede dividir un sistema en tres partes. Como se muestra en la figura 3.2 esta división puede dar en: (a) *presentación*, el cual se refiere al procesamiento que ocurre con el fin de exhibir información, si la pantalla es basada en carácter o gráfica, si es una PC o una terminal; (b) *lógica de la aplicación*, el cual se refiere al procesamiento asociado con la ejecución de la aplicación; y (c) *manejo de datos* el cual cubre el procesamiento asociado con bases de datos y sistemas de administración de base de datos.

En lo que respecta a la red esta adquiere la capacidad de un sistema distribuido, tanto como sus servicios de aplicación distribuidos crezcan. Así, estos componentes son enlazados por una diversidad de servicios de comunicación. En la figura 3.3 se ilustra una posible estructura de un nodo de red presentado en OSNA (Open System Network Architecture). OSNA es un modelo evolucionado de Arquitectura de Sistemas en Red (SNA).

System Network Architecture) de IBM. Este es un modelo abstracto, contra el cual los sistemas distribuidos pueden ser examinados. La estructura de sistemas distribuidos pueden soportar tres conjuntos de servicios:

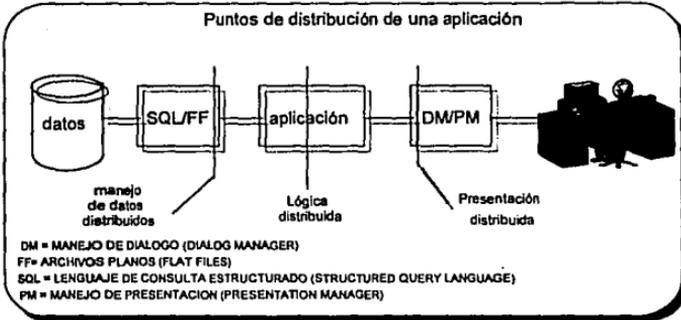


Figura 3.2 Componentes de una aplicación

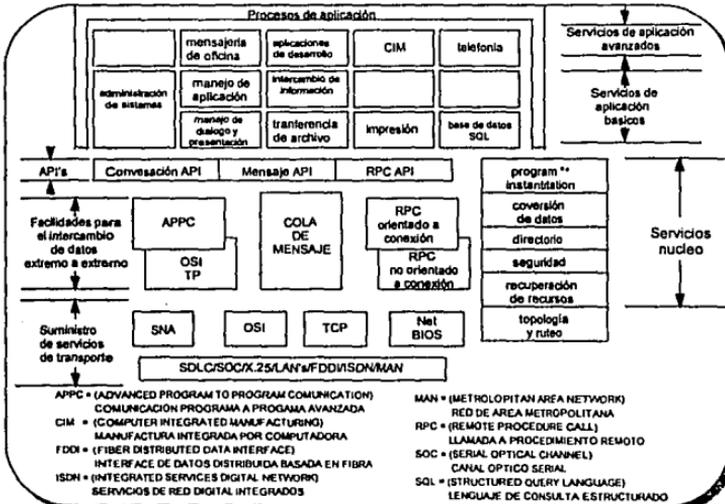


Figura 3.3 Estructura de un nodo de red OSNA

1. Un núcleo de servicios de aplicación que definen el propósito general y esencial de un sistema operativo de red.

2. Un conjunto básico, el cual incluye otros muy comunes servicios de aplicación usados
3. Servicios de aplicación avanzado (más especializado)

La selección de servicios que es apropiado en una instalación en particular, es por supuesto, variables en cada empresa. Sin embargo muchas redes pueden crecer con un conjunto consistente de los siguientes servicios distribuidos:

- SopORTE núcleo del sistema
 1. Seguridad
 2. Recuperación de recursos
 3. Directorio
 4. Conversión de datos
- SopORTE de aplicación básica
 1. Administración de sistema
 2. Transferencia de archivos
 3. Base de datos SQL y consulta
 4. Manejador de aplicaciones
 5. Intercambio de información multimedia
 6. Impresión
 7. Manejo de diálogo/presentación
- En adición servicios de aplicación avanzados o especializados pueden por ejemplo incluir servicios para:
 1. Almacenamiento y envío de mensajes
 2. Servios de aplicación telefónico
 3. Ambiente de desarrollo de aplicaciones
 4. aplicaciones de manufactura integrada por computadora

3.1.4. DISTRIBUCION DE LA ARQUITECTURA CLIENTE/SERVIDOR

Como ya se menciona, tres son los elementos a distribuir entre la arquitectura Cliente/Servidor: el manejo de datos, la aplicación y la presentación. En la figura 3.4, cada uno de estos tres tipos de procesamiento puede ser distribuido, en distintos grados, entre el sistema de anfitrión y el escritorio dependiendo de cual configuración se ajuste mejor encuentra los requerimientos individuales de una organización.

Lo que es importante recordar es que el "anfitrión" no es sinónimo de "servidor", ni el "escritorio" es sinónimo de "Cliente"; en muchos ejemplos la PC de escritorio o la terminal pueden actuar como servidor respondiendo a solicitudes de cliente desde el anfitrión.

Arquitectura 1: Presentación Distribuida

En este modelo, toda la base de datos y procesamiento de aplicación ocurre en el anfitrión. El procesamiento de presentación, sin embargo, se divide entre la computadora anfitrión y la terminal de escritorio o PC. Este es un caso donde la terminal de escritorio o PC actúa de servidor en vez de cliente: la aplicación, localizada básicamente en el anfitrión, envía solicitudes (tal como secuencias de dibujo con caracteres o direccionamiento del cursor) a la terminal de escritorio o PC la cual responde a esas solicitudes (como un servidor) dibujando el carácter solicitado o moviendo el cursor.

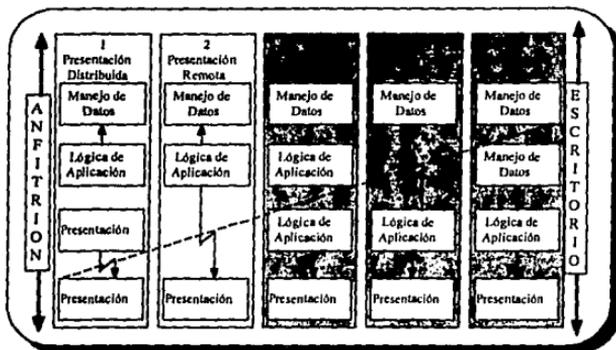


Figura 3.4 Distribución de cinco configuraciones cliente/servidor. La línea punteada indica la separación del anfitrión y el escritorio por medio de la red

Esta arquitectura representa la forma más rudimentaria de procesamiento distribuido y frecuentemente ni siquiera es percibido como componente de una configuración cliente/servidor.

Esta configuración es apropiada para organizaciones que desean control centralizado de la base de datos y procesamiento de aplicación el cual, en correspondencia, este simplifica la administración y mantenimiento de sistemas. Por su naturaleza, pocas ganancias serán logradas en términos del rendimiento del sistema o tiempo de respuesta porque el volumen de procesamiento está todavía situado en el anfitrión.

Arquitectura 2: Presentación Remota

En la configuración de Presentación Remota, virtualmente todo el procesamiento de presentación es descargado del anfitrión y manejado por la PC de escritorio. Como en la arquitectura previa, el anfitrión, actúa como un cliente, envía solicitudes de presentación a la computadora de escritorio. La PC, actuando como un servidor de presentación, interpreta estas solicitudes y determina como los datos serán exhibidos.

La diferencia significativa entre esta configuración y la Presentación Distribuida es la inclusión de un *servidor de objeto remoto*. En esta configuración, virtualmente toda la lógica de presentación está situada en el escritorio, también ofrece un grado mayor de procesamiento distribuido, reduciendo la carga en el anfitrión.

Además, al situar el procesamiento de presentación gráfica en el escritorio, la interfaz de presentación entre el anfitrión y cliente pueden ser más sencilla y robusta (ya que menos información es pasada entre cliente y servidor).

La Presentación Remota ofrece a los gerentes de sistemas, un modo eficiente de manejar la creciente demandas de procesamiento de interfaces gráficas, las cuales generalmente consumen más recurso que las de software basado en carácter. Al utilizar el poder del PC de escritorio para almacenar y procesar información localmente, la carga de procesamiento en el anfitrión puede ser reducida, permitido a los gerentes de sistemas capitalizar el uso de las amigables GUI's sin disminuir el rendimiento del sistema

Arquitectura 3 : Función Distribuida

Función Distribuida -Donde la lógica de aplicación es dividida entre el anfitrión y el escritorio- es la implementación más compleja de computo cliente/servidor. Requiere que secciones discretas de la lógica de aplicación sean separadas y situada en aquellas áreas que pueden ofrecer un máximo de mejoramiento en el rendimiento del sistema sin comprometer la funcionalidad de la aplicación.

La configuración cliente/servidor de Función Distribuida mantiene la promesa de mayor flexibilidad en la distribución de procesamiento en el sistema. Al cargar algo del procesamiento de la aplicación al escritorio, la carga en el anfitrión es reducida y el procesamiento requerido por los recursos puede ser asignado más eficientemente en el sistema. Mientras que es teóricamente posible diseñar una aplicación para esta configuración, la complejidad significativa asociada con su complejidad lo hace extremadamente difícil en la práctica. En casos aislados sin embargo, donde las aplicaciones no necesita tener acceso común (o compartido) a datos en memoria, este método podría ser de beneficio.

Arquitectura 4 :Manejo de Datos Remota

La configuración de Manejo de Datos Remota representa la definición más comúnmente encontrada de computo cliente/servidor. Una PC de escritorio que maneja la aplicación y la lógica de presentación está conectada vía red a un anfitrión central, donde los datos son almacenado.

Mientras que esta configuración hace la administración y mantenimiento del sistema más difícil (más que las arquitecturas de Presentación Remota o Distribuida) permite a los gerentes de sistemas cargar la gran mayoría de proceso al escritorio de ese modo reduciendo significativamente el tráfico de red y la carga de procesamiento en el anfitrión.

Arquitectura 5 : Base de datos Distribuida

La configuración cliente/servidor de Base de datos Distribuida mantiene la promesa de un sistema de más bajo gasto y de amplio procesamiento del sistema al situar un conjunto de datos en la PC de escritorio.

El beneficio de esta arquitectura es que, en adición a la carga de procesamiento de aplicación y presentación desde el anfitrión, la ganancia en rendimiento del sistema puede ser maximizada al situar archivos de datos en la PC de escritorio. En su forma más simple, esto podría ser archivos-fuente no compartidos (tal

como registros de sólo-lectura), pero es difícil mantener la misma base de datos en el anfitrión y en el escritorio. Así, al almacenando archivos de datos localmente, el tráfico de red es reducido y los tiempos de respuesta podrían ser más rápidos.

3.2 CLIENTE/SERVIDOR

En la práctica el termino cliente se refiere a los sistemas que al ejecutar un proceso solicitan servicios.

Figura. 3.5. El termino servidor se refiere a los sistemas que al ejecutar un proceso ofrecen servicios.

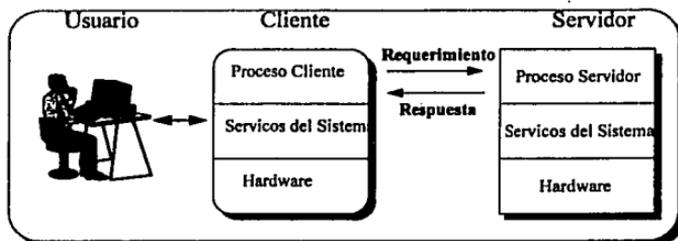


Figura. 3.5 Sistema Cliente/Servidor

Hay tres niveles tanto en el cliente como en el servidor, como se muestra en la figura 3.6. El primer nivel esta compuesto por el hardware del sistema. El siguiente nivel es la capa de servicios del sistema, el cual abarca el sistema operativo de red. La capa de servicios del sistema incluye todo el software usado por la aplicación para controlar el hardware. El nivel más alto es la capa de aplicación que consiste en los procesos del cliente o del servidor y algunos otros procesos de aplicación. Se hace notar que en un sistema cliente/servidor la aplicación abarca tanto al cliente como al servidor. Es importante reconocer que el paradigma cliente/servidor puede ocurrir también en el nivel de servicios del sistema.

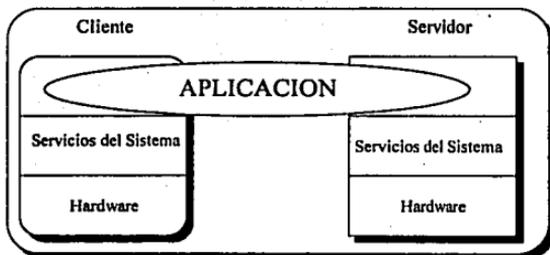


Figura 3.6 Aplicación cliente/servidor

La diferencia primaria entre implementaciones en la capa de aplicación y la de servicios del sistema es relativo a la localización de los componentes en la jerarquía. En una aplicación cliente/servidor en la capa de

aplicación, el "usuario" es una persona y el "cliente" y "servidor" llevan a cabo la aplicación. En implementaciones en la capa de servicios del sistema, El "usuario" es típicamente un programa de aplicación, el "cliente" es un redirector del procesos y el "servidor" es el que da una respuesta al proceso.

Se examinarán las características principales del modelo cliente/servidor en mayor detalle.

3.2.1. ATRIBUTOS Y COMPONENTES DEL CLIENTE

El proceso del cliente es proactivo, emitiendo solicitudes al servidor. Esta típicamente dedicado a sus usuarios de sesión y es quien comienza y termina con la sesión. Un cliente puede interactuar con un sólo servidor o con múltiples servidores para realizar su trabajo.

En el nivel de aplicación, el cliente es el responsable de mantener y procesar el diálogo completo con el usuario. Esto incluye la ejecución completa de lo siguiente:

- Manejo de pantalla
- Interpretación de menús y comandos
- Validación y registro de datos
- Procesamiento de ayuda
- Recuperación de errores

En aplicaciones gráficas, además se incluye

- Manejo de ventanas
- Entradas de mouse y teclado
- Control de cajas de diálogo
- Manejo de sonido y vídeo (en aplicaciones de multimedia)

En la práctica, el cliente no puede ejecutar toda la validación de datos siempre. Sin embargo, este puede usualmente manejar todos los datos previos a las actividades, edición individual, mezcla de campos y edición de muchas tablas para manejar datos (tal como validación de código usado).

En la capa de servicios del sistema, el cliente es el responsable de detectar e interceptar solicitudes de servicio desde la aplicación y redirigirlas de conformidad. El redirector no debe simplemente rutear todas las interacciones del "usuario" al servidor si es parte de un sistema cliente/servidor verdadero.

COMPONENTES DE SERVICIO EN EL CLIENTE

En el modelo cliente/servidor, el cliente es primeramente un consumidor de servicios proporcionados por uno o más servidores. El modelo proporciona una clara separación de funciones basado en la idea de que el servidor actúa como suministrador de servicios respondiendo a las solicitudes del cliente. El cliente solicita los servicios del servidor a través del *redirector* del sistema operativo de red. Este servicio intercepta las llamadas del sistema operativo local y las redirecciona al sistema operativo del servidor. Algunos de los servicios disponibles en el cliente son:

- **Servicio de fax/impresión.** El NOS (Network Operative System) habilita al cliente para generar servicios de impresión. Estos son redirigidos por el software redirector del NOS y manejados por el administrador de la cola de impresión. El cliente puede visualizar el estatus de la cola de impresión en cualquier momento.
- **Servicio de ventanas.** En las aplicaciones, usando GUI's (Graphic Unit Interface), colocan datos en la pantalla virtual y los servicios de ventanas manejan la colocación y manipulación de aplicaciones. El cliente puede tener varias ventanas abiertas al mismo tiempo la capacidad de activar, visualizar, mover, cambiar tamaño u ocultar una ventana es proporcionada por los servicios del software de la GUI.
- **Servicio de inicialización remoto.** Algunas aplicaciones operan bien sobre estaciones de trabajo sin discos de almacenamiento local, las terminales-X y estaciones de trabajo son ejemplos. La estación de trabajo cliente debe proporcionar suficiente software dentro la memoria de solo lectura programable-borrable (E_PROM) para comenzar el proceso de carga inicial de programas. E_PROM incluye las entrada/salidas básicas del sistema (BIOS).
- **Servicios de utilidad.** El sistema operativo local proporciona funciones locales como copia, edición, movimiento, comparación y ayuda que se ejecuta en la estación de trabajo cliente.
- **Servicio de mensajes.** Los mensajes pueden ser enviados y recibidos en forma sincronizada a y desde la red. El servicio de mensajes proporciona los servicios de almacenamiento de paso, planeación y arbitraje para soportar estas funciones.
- **Servicios de aplicación.** En adición a los servicios remotos de ejecución que proporciona NOS, aplicaciones especializadas usan sus propios API's incrustados en un RPC para invocar servicios especializados desde un servidor remoto.
- **Servicios de base de datos.** Un manejador de base de datos (DBMS) es diseñado para almacenar y recuperar una gran cantidad de información y para proporcionar aplicaciones con acceso concurrente a la información almacenada en la base de datos en forma de objetos.

3.2.2. ATRIBUTOS Y COMPONENTES DEL SERVIDOR

El servidor de procesos es reactivo, iniciando un proceso a la llegada de las solicitudes de sus clientes. Un servidor de procesos usualmente siempre esta activo, proporcionando servicios a muchos clientes.

Un servidor tiene funciones específicas: Ejecuta un conjunto predefinido de transacciones relacionadas funcionalmente. Existen frecuentemente múltiples servidores que proporcionan funciones necesarias a su comunidad de clientes.

Un servidor ejecuta toda la lógica necesaria para procesar una transacción y puede interactuar con otros servidores. Si un cliente trabaja con múltiples servidores, es responsabilidad del cliente invocarlos cuando lo

requiera. Los procesos con el servidor típicamente incluyen acceso, almacenamiento y organización de datos compartidos; actualización de datos previamente almacenados; y el manejo de otros recursos compartidos.

En la actualidad existen servidores de presentación en pantalla que ejecutan X-windows en las terminales como se muestra en la figura 3.7. El protocolo X-Windows es cliente/servidor completamente en la capa de servicios del sistema. El cliente y el servidor pueden ejecutarse en la misma máquina o en máquinas separadas.

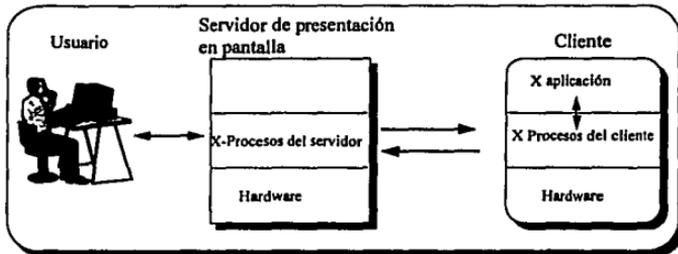


Figura 3.7 Implementación en Cliente/Servidor en X-Windows

Se observa que el servidor de presentación es reactivo al cliente X y al usuario. Además el servidor de presentación puede desplegar solicitudes emitidas desde múltiples clientes sin tener en cuenta su localización. Esta es la única característica que diferencia al X-windows de redes no orientadas a ambientes Windows.

En el nivel de aplicación, la aplicación X puede ser o no cliente/servidor. Por ejemplo si la aplicación es un procesador de texto se ejecutara completamente en el cliente. Alternativamente, la aplicación puede ser parte de una aplicación grande en el cliente interactuando con un servidor de bases de datos, por ejemplo, en forma cliente/servidor. Esto se muestra en la figura 3.8.

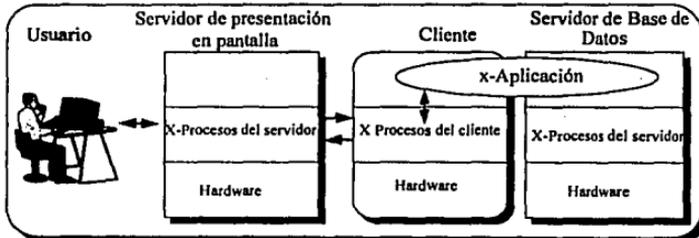


Figura 3.8 Aplicación en Cliente/Servidor en X-windows

Si el cliente mostrado en la figura 3.8 soporta múltiples aplicaciones, puede ser creado un interesante efecto llamado **servidor de acceso**. En el nivel de servicios del sistema, el sistema cliente ejecuta simplemente un cliente para el servidor de presentación en pantallas. En el nivel de aplicación, sin embargo, el sistema

cliente aparenta ser un servidor que ofrece acceso al sistema de presentación en pantalla para una variedad de aplicaciones. Sin embargo, en realidad, el sistema cliente corre sólo la porción (código) del cliente correspondiente a esa aplicación. El sistema cliente no es un servidor del todo ya que no corresponde al proceso del cliente que está corriendo en el nivel de aplicación sobre el servidor de presentación en pantalla. El efecto es creado porque el servidor de presentación en pantalla usa al cliente como un punto de acceso remoto para estas aplicaciones.

El efecto del servidor de acceso tiene usos prácticos. Por ejemplo, si se necesita desplegar varias aplicaciones que no pueden correr físicamente en la terminal. Esto debido a la memoria, sistema operativo o limitaciones del CPU. Esto es ilustrado en la figura 3.9. las terminales pueden ser configuradas como servidores de presentación en pantalla, accediendo a grandes servidores de acceso en redes LAN para trabajo en grupo.

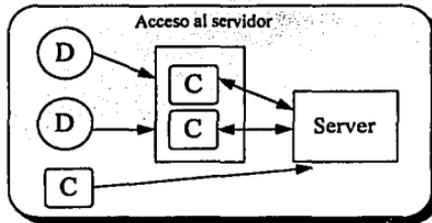


Figura 3.9 Uso práctico para acceder al servidor

La mayoría de los servidores que proporcionan acceso a datos compartidos están en el nivel de aplicación. Una excepción son los servidores de archivo, los cuales están en el nivel de servicios del sistema. En el nivel de aplicación los servidores de aplicaciones usualmente proporcionan acceso a tipos específicos de datos: servidores de personal, de entrada de ordenes, imágenes, texto, etc.

FUNCIONES ESPECIFICAS DEL SERVIDOR

- **Solicitudes de procesamiento.** Las solicitudes son emitidas desde un cliente al servicio de software del NOS residente en el cliente. Estos servicios le dan el formato de RPC apropiado a la solicitud y emite la solicitud a la capa de aplicación del protocolo del cliente. Esta solicitud es recibida por la capa de aplicación del protocolo en el servidor.
- **Servicios de seguridad.** A cada usuario se le solicitará un login y un password. Como el password puede llegar a ser visible por usuarios no autorizados. El servidor debe permitir cambiar el password regularmente.
- **Servicios de fax/impresión e imagen.** Impresoras de alta calidad, estaciones de trabajo con servicios de fax y graficadores son candidatos naturales para agregarse a un servidor compartido. El servidor puede

aceptar entradas desde muchos clientes, colocarlos en la cola de acuerdo a la prioridad de la solicitud y canalizar la solicitud cuando el dispositivo este disponible.

Los faxes que llegan pueden ser formados en la cola del servidor y transmitidos al cliente apropiado. De acuerdo a técnicas de manejo del flujo de trabajo, las imágenes pueden ser capturadas y distribuidas al cliente apropiado desde el servidor un imágenes: en el modelo cliente/servidor, la cola de trabajo es mantenida y supervisada por un supervisor basado en algoritmos que por default determinan como distribuir la cola de trabajo.

Servicios de archivo. Los servidores de disco fueron los primeros intentos de dispositivos de almacenamiento compartido. Estos servidores de disco simplemente proporcionaban acceso al disco en un nivel bajo para múltiples estaciones de trabajo. Los servidores de archivo fueron la siguiente generación de los servidores de disco. Como su nombre lo indica se diseñaron para proporcionar servicios de archivo a múltiples usuarios.

Los servicios de archivo manejan el acceso a directorios virtuales, archivos localizados en el cliente y almacenamiento permanente en el servidor. Estos servicios son proporcionados a través del software redirector implementado como parte del sistema operativo del cliente. En la típica implementación el software, datos compartidos y respaldos almacenados en discos, cintas y dispositivos de almacenamiento óptico, que son manejados por el servidor de archivos.

Network File Systems (Sistemas de Archivos en Red). Un problema asociado con los archivos compartidos en redes es que la arquitectura básica de las computadoras puede ser muy diferente. Esta diferencia se extiende desde el orden de los bytes de datos hasta como la computadora representa números de punto flotante y texto.

El System file network (NFS) fue creado para aliviar este problema. El NFS fue desarrollado por Sun Microsystem en 1984 para ser sistema operativo de red y protocolo de transporte independiente de la máquina. NFS proporciona uso remoto y transparencia del sistemas de archivo compartido en redes locales. NFS lleva a cabo esta independencia a través de dos protocolos en capa, llamadas a procedimiento remoto (RPC) y representación externa de datos (XDR). RPC maneja el intercambio de comandos y datos con un servidor remoto, permitiendo a los procedimientos ser ejecutados en una máquina remota como si fueran llamadas a procedimientos de software local. Las primitivas de RPC están construidas arriba de XDR, la cual proporciona un método independiente de la máquina para representación de datos y elimina dependencias tales como ordenamiento de bytes y formato de punto flotante.

En el nivel base de NFS se ejecutan dos funciones primarias: juntas exportan y montan el sistema de archivo. Una computadora exporta un sistema de archivo, diciéndole a todas las demás computadoras de la red que tiene este sistema de archivo en particular disponible para su uso. El montaje consiste en unir

un sistema de archivo a un lugar específico en el árbol de archivo. En la figura 3.10 se ilustra la estructura de NFS.

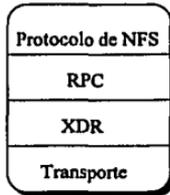


Figura 3.10 Capas del Sistema NFS de SUN

- **Servicios de bases de datos.** Con la expansión de sistema cliente/servidor, los servidores de bases de datos han llegado a ser una parte importante del ambiente de redes locales. Los servidores de bases de datos tienen dos configuraciones. La primera configuración es stand-alone, donde el servidor de base de datos permanece sobre una estación de trabajo dedicada. En la segunda y nueva configuración, el servidor de base de datos reside sobre un servidor de archivo. En esta situación el servidor de bases de datos es dinámicamente enlazado al sistema operativo de la red. Estrechando la integración del motor de base de datos y el sistema operativo tiene una mayor de seguridad y eficiencia.

3.3 COMUNICACIONES

La teoría de programación concurrente esta basada en la noción de que los procesos de comunicación se ejecutan en paralelo, dejando pasar mensajes a través de canales de comunicación compartidos. Se utilizará el termino interprocesos de comunicación (IPC) para referirnos a las técnicas de transferencia de mensajes en general.

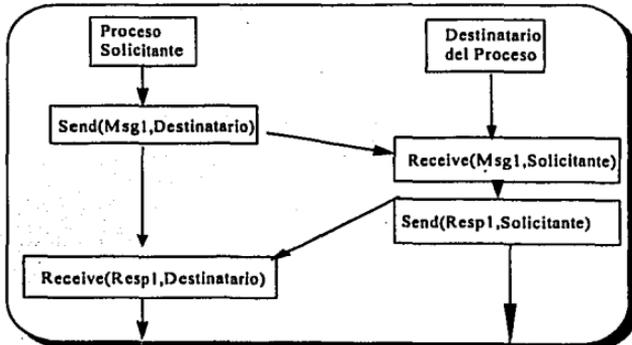


Figura 3.11 Esquema general de la transferencia de mensajes

Para enviar un mensaje, un proceso ejecuta `send(mensaje, destinatario)` llamando al sistema IPC. Para recibir el mensaje, un proceso ejecuta `receive(mensaje, remitente)`. Esto es mostrado en la figura 3.11.

Pero en la transferencia de mensajes basados en cliente/servidor, un servidor necesita ser capaz de recibir mensajes desde varios clientes, necesitando un mecanismo general, llamado puerto. En la figura 3.12., un puerto es esencialmente como un buzón, al cual, cualquier proceso puede enviar un mensaje pero solo un proceso puede recibirse, los mensajes son recibidos en el orden en el cual fueron puestos en el buzón. Este proceso es mostrado en la figura 3.13.

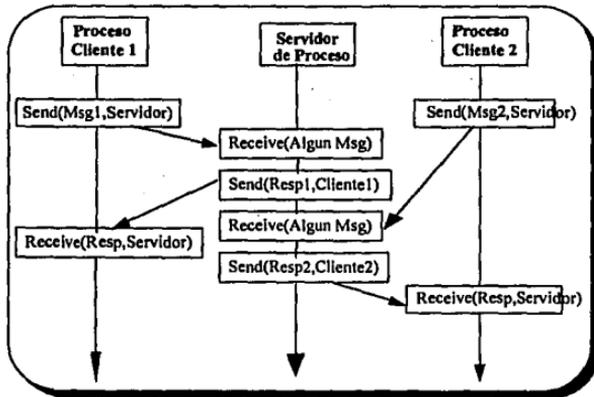


Figura 3.12 Recepción tipo buzón

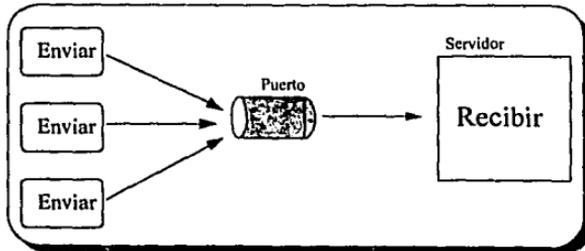


Figura 3.13 Ejemplo de puerto

Las tres principales cuestiones discutidas en el modelo de comunicaciones envío/recepción son:

- Tipo de conexión entre emisor y receptor
- Entrega de mensajes
- Actividades de sincronización

3.3.1. ASPECTOS DE CONEXIÓN

Hay dos tipos de conexión usadas para comunicaciones. El primer tipo *sin conexión*, donde cada mensaje encuentra su propio camino a su destino. Cada mensaje es independiente de otro mensaje y puede seguir diferentes rutas para llegar a su destino, ya que cada mensaje debe tener la información completa de la dirección donde será entregado. Los mensajes en una comunicación sin conexión son llamados *datagramas*. Esta forma de comunicación es también conocida como *conmutación de paquetes*.

El otro tipo de comunicación es el *orientado a conexión*, en donde un *circuito* es establecido antes de que la comunicación ocurra, también conocida como *circuito conmutado*. Una vez que el circuito es conectado, cada mensaje sigue una secuencia y siempre es ruteado a lo largo del mismo circuito. Así, cada mensaje sólo necesita un identificador de circuito para ser ruteado a su destino. La llegada de cada mensaje es confirmada y, si es necesario, es usado un control de flujo para regular la velocidad con la cual los mensajes son enviados. Cuando la comunicación termina, el circuito es explícitamente deshecho. Dependiendo del protocolo de red usado, establecer un circuito puede resultar en una ruta dedicada a cada circuito (ej. un circuito virtual permanente), o una ruta compartida entre varios circuitos lógicos. Los mensajes en una comunicación orientada a conexión son llamados *flujo de datos*. La tabla 3.1 resume las diferencias entre las comunicaciones sin conexión y orientadas a conexión. La mayoría de los principales protocolos soportan ambos tipos de comunicación. Por ejemplo, NetBIOS, Internet, Novell, OSI y AppleTalk.

Un circuito en dos sentidos es llamado *duplex* y permite en ambos lados enviar mensajes simultáneamente. Un circuito en un sólo sentido es llamado *half-duplex* y necesita que ambos lados estén de acuerdo en quien va a enviar un mensaje y en que momento. Un circuito *simplex* es otro tipo de un circuito en un sentido. Se hace notar que los datagramas no son ni duplex o half-duplex, no hay conexión.

Características	Orientado a conexión	Sin conexión
Tipo de mensaje	Flujo de datos	Datagrama
ruteo	Estático	Dinámico
Direccionamiento del mensaje	Dirección del destino para establecer el circuito; después sólo una id del circuito.	Búsqueda en todos los nodos o dirección del destino para un nodo específico
Confiabilidad	Secuenciado, control de error, control de flujo y entrega garantizada	Sin garantías, los mensajes pueden perderse o llegar fuera de orden
Opciones	Puede ser negociada durante el establecimiento	No definida
Sincronización	Explícita	Implícita
Gastos	Establecimiento del circuito y baja	Ruteo del mensaje

Tabla 3.1 Características de tipos de conexión

3.3.2. ASPECTOS DE DIRECCIONAMIENTO

VINCULACION ESTATICA

En una red, es utilizado el par (dirección anfitrión, id-proceso), usando un esquema de nombre directo, el cliente debe saber la dirección física sobre la red del nodo en el que los procesos del servidor corren. Ya que el servidor puede tener varios clientes, el cliente debe incluir su propia dirección junto con el mensaje y así el servidor sepa donde enviar los resultados.

Pocos protocolos utilizan la técnica de vinculación estática. La dificultad radica en que el cliente usualmente no sabe la dirección física del servidor en la red.

IDENTIFICACION DE NOMBRES

En cualquier red, el nombre "A" puede ser usado para llevar al identificador "B", dentro de un contexto específico. El nombre "B" puede ser usado en el siguiente nivel hacia abajo para llevar al identificador "C" y así sucesivamente. Como se muestra en la figura 3.14., el manejo de mensajes en un ambiente TCP/IP usualmente involucra tres contextos de nombres: socket (conexión), network(red) y link layers (enlace de capas).

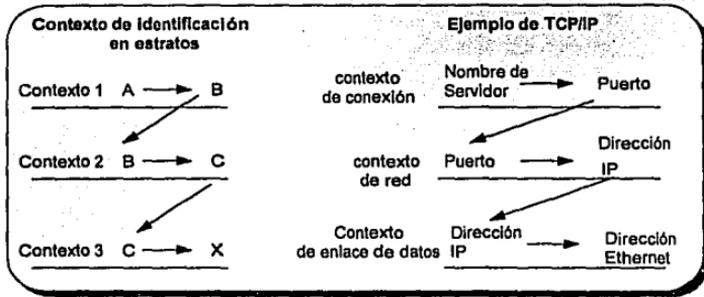


Figura 3.14 Contexto de identificación en estratos

NOMBRES DE AREA GLOBAL.

Un nombre global debe ser único en el contexto en el cual será utilizado. Un identificador de proceso es único en el contexto de una computadora, pero no en una red de computadoras, llevando una identificación del anfitrión consigo el identificador de proceso hace el nombre más global. En una red grande, el identificador del anfitrión puede no ser único y un identificador de una subred puede ser necesario para hacer el nombre más global y así sucesivamente. Este es un ejemplo de como la concatenación jerárquica puede ser usada para construir nombres de áreas global. Cada nombre es concatenado con su contexto para construir un nombre global. EJ.

red + subred + anfitrión + id-proceso + nombre

Otra forma de construir un nombre de área globales es usando un formato de asignación de nombres uniforme y particionar el rango de valores que el nombre puede tener. Por ejemplo puede acordarse que todos los anfitriones usen cuatro dígitos para identificar sus procesadores-clientes, donde el anfitrión 1 usara un identificador de procesadores-cliente en el rango 0000-0999, el anfitrión 2 usara del 1000-1999, etc. Ambas técnicas pueden ser combinadas en la creación de nombres de área global.

Un ejemplo práctico de esto es el esquema de asignación de nombres por dominio Internet. El formato básico de un nombre internet es *usuario@subdominio.dominio*. El dominio es una colección grande de computadoras en red que permiten el acceso transparente a la ubicación de algunos datos dentro del dominio. Cada dominio se usa también para aspectos de administración y seguridad. El nombre de área del dominio es particionado por un sufijo estándar que indica ya sea el tipo de organización (COM, EDU, GOV, MIL, etc.) o su localización geográfica (MEX, DF, TLAX). EJ. MYANFIT.VALERIANO@UNAM.DF.

VINCULACION DINAMICA VIA CONVENCION.

Hay varias formas de hacer la vinculación dinámica. El método más simple usa una configuración convenida para establecer como el nombre será localizado de acuerdo a una dirección en la red. Tal convención usa una variable de ambiente para localizar el nombre global de acuerdo a una dirección en la red. La variable puede ser inicializada en la configuración de arranque (ej. MISERVIDOR=123456). Esto no sólo desperdicia memoria, sino además estando en uso la variable de ambiente esta impedido su uso por otro proceso ejecutándose durante la misma sesión.

Otra forma de hacer una vinculación dinámica es usar un archivo de configuración que contenga nombres globales y su dirección en la red, todos los procesos de clientes en el mismo nodo pueden consultar este archivo en orden para localizar a su servidor

VINCULACION DINAMICA VIA BUSQUEDA (BROADCAST).

Ya que la configuración de convenciones son restrictivas, la mayoría de los protocolos proporcionan un servicio de resolución de nombres en forma dinámica. Por ejemplo, en el protocolo NetBIOS, una tabla residente en memoria es usada en lugar del archivo de configuración, esto es ilustrado en la figura 3.15, los procesos del cliente pueden explícitamente agregar nombres globales a la tabla antes de que puedan usarla en un envío cuando un nombre global es agregado(1), NetBIOS chequea que el nombre no este ya en uso y entonces transmite este nombre a todos los nodos en la red (2). El servidor al que corresponde este nombre global responde con su dirección en la red (3). NetBIOS actualiza sus tablas y utiliza la dirección retornada en lugar del nombre global en todas las demás comunicaciones (4-5).

Una forma diferente de hacer una vinculación dinámica, en vez de una tabla de nombre global, es el transmitir siempre el nombre global a todos los nodos en la red. El servidor puede responder cuando capte su nombre transmitido. Aunque las transmisiones continuas consumen ancho de banda, esto puede ser eficientemente implementado en protocolos que entregan todos los paquetes a todos los nodos de alguna forma.

Por ejemplo, en una topología Token ring un paquete viajara pasando por cada nodo en el anillo hasta que sea removido por su nodo destino. La principal limitación del método es que los ruteadores de la red no propagaran la transmisión de paquetes a otras subredes.

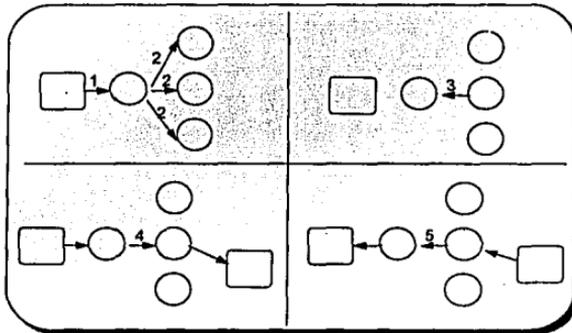


Figura 3.15 Identificación del nombre con NetBIOS

Si solo unos pocos mensajes serán enviados, frecuentemente tiene sentido combinar paquetes que llevan la dirección con paquetes que llevan mensajes. Esto es, en lugar de enviar la transmisión de un paquete para localizar un servidor, el mensaje mismo puede estar encapsulado con el paquete transmitido. Por ejemplo, los protocolos NetBIOS e Internet ofrecen un servicio de datagramas que permite la vinculación tipo búsqueda. El resultado es un manejo de mensajes muy rápido, ya que los datagramas eliminan la necesidad de cualquier contactación para asegurar que el destinatario ha recibido el mensaje. Mientras frecuentemente el uso de este estilo de manejo de mensaje congestiona la red, la transmisión de datagramas usualmente viaja dos veces más rápido que la más confiable alternativa de manejo de mensajes.

VINCULACION DINAMICA VIA NOMBRE DE SERVIDOR

Por último, un nombre de servidor explícito puede ser usado para llevar a nombres globales. En este esquema, un proceso cliente interactua con un nombre de servidor, el cual retorna la dirección en la red del nombre global. El cliente puede entonces comunicarse directamente con el servidor proyectado, como se muestra en la figura 3.16

Esto introduce un problema recursivo: ¿como localiza el cliente el nombre del servidor?. La mejor forma es transmitir el nombre del servidor buscado y un nombre de servidor. La conversación con el nombre del servidor es típicamente corta, ya que el cliente solamente quiere saber la dirección del servidor buscado. Los datagramas son ideales en esta situación. El cliente transmite "¿donde esta mi servidor?" y el nombre del servidor responde con la respuesta. Si el datagrama esta perdido, el cliente puede esperar un tiempo por la respuesta y retransmitir la solicitud.

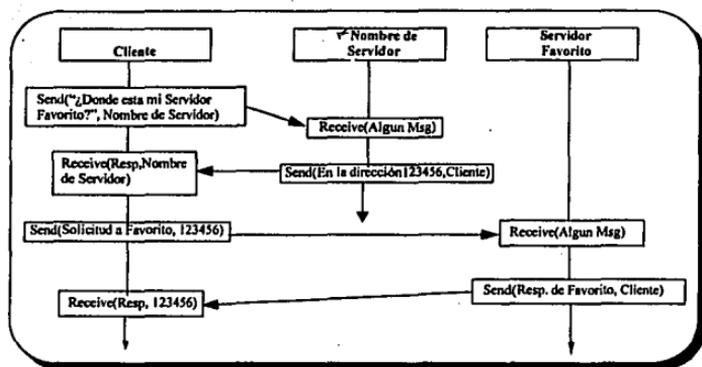


Figura 3.16 Nombre de Servidor genérico

Un nombre de servidor individual puede ser un punto de falla. Además si el nombre del servidor está de baja, las continuas retransmisiones de los clientes rápidamente consumen el ancho de banda de la red. Sin embargo, se puede superar este problema con la transmisión de múltiples nombres de servidores y limitando el número de reintentos del cliente. El múltiple nombre de servidores fácilmente ajusta en el modelo de transmisión. Por ejemplo, si dos nombres de servidores existen ambos contestaran a la misma transmisión del datagrama. Ya que el cliente captará solo una respuesta, la respuesta del otro servidor será descartada.

Una variación es el uso de nodos adyacentes como nombre de servidores para sus vecinos. Por ejemplo, en un protocolo basado en Token-ring los vecinos hacia abajo en el flujo pueden interceptar la transmisión si conocen la respuesta y reemplazar el mensaje transmitido con un mensaje de respuesta. Esta variación trabaja bien en grupos de trabajo homogéneos donde los nodos adyacentes corren la misma carga de trabajo. Por lo tanto ellos conocen siempre la respuesta para un nuevo nodo sobre el flujo de la carga de trabajo.

El método del nombre de servidor trabaja bien especialmente en situaciones donde el servidor proyectado es remoto y no puede responder a la transmisión el mismo. Por ejemplo, si el servidor está en otro sitio la transmisión de mensajes no puede cruzar la WAN. Un nombre de servidor local puede responder a la transmisión en nombre del servidor proyectado suministrando la más reciente dirección a el cliente. En redes grandes, el servidor proyectado está frecuentemente alejado y un nombre de servidor local es importante. Sin embargo se debe tener cuidado en escoger una implementación eficiente, los protocolos de red parlan-chines, como son Apple Talk o NetBIOS, hacen libre uso de la transmisión de mensajes para mantener el nombre de sus servicios actualizados, esto consume ancho de banda indispensable en una red grande.

3.3.3. ASPECTOS DE SINCRONIZACION.

Las características de sincronización de los mensajes de un protocolo gobiernan si un proceso deja de correr cuando ejecuta un envío o una recepción. Un protocolo sin bloqueo permite al proceso continuar corriendo, mientras que un protocolo con bloqueo suspende el proceso hasta que la operación se complete exitosamente. Un envío exitoso se da cuando el sistema de mensajes fue capaz de alojar este en un buffer interno para retener el mensaje enviado. Una operación de envío bloqueada detendrá al emisor hasta que el sistema operativo aloje el mensaje en un buffer. Un envío sin bloqueo regresará un código de error al emisor si no hay buffers libres, permitiendo al procesador hacer otro trabajo. Una operación de recepción bloqueada parará el proceso de recepción hasta que el mensaje llegue, mientras que una recepción sin bloqueo simplemente indica que el mensaje no está presente.

Si todas las operaciones que son enviadas están implementadas con bloqueo, el contenido del mensaje reflejara el estado actual de lo proceso enviado. En el punto en que el mensaje es entregado, el procesador emisor y receptor están sincronizados.

Si la operación de send no esta sin bloqueo, el emisor puede adelantarse del receptor, por lo tanto cuando el mensaje es entregado a el receptor su contenido no refleja el estado actual del emisor.

Se hace notar que en las operaciones de receive sin bloqueo no afecta si el mensaje pasado es sincronizado.

Es fácil programar usando mensajes sincronizados, ya que la lógica de programación es simple. El uso de mensajes asíncronos es empleado en aplicaciones de tiempo real donde el emisor no puede tomar tiempo para esperar.

3.4 LLAMADAS A PROCEDIMIENTOS REMOTOS

Un RPC es un procedimiento que se ejecuta en otra máquina diferente a la que hizo la invocación del procedimiento, es decir, el cliente no ejecuta el procedimiento, solo lo invoca en el servidor. Cuando un proceso ejecuta un RPC y se suspende; los parámetros de la llamada son enviados a donde se encuentra el procedimiento remoto; y el proceso es ejecutado ahí, cuando el procedimiento remoto se completa, sus resultados son enviados de regreso a través de la red y el proceso llamado reanuda el procesamiento como si estuviera regresando de una llamada a un procedimiento local. Esto se ilustra en la figura 3.17

Se hace notar que al ejecutar un procedimiento remotamente tiene la misma semántica que ejecutar el mismo procedimiento localmente. Para realizar esto, el ambiente de la llamada (pila de proceso y variables globales) deben ser transferidas al procesador destino, a menos que haya restricciones en el tipo de procedimiento llamado. Por ejemplo, cuando un procedimiento remoto pasa un puntero a una estructura de datos grande o tiene referencia a una variable global, a menos que el ambiente del proceso de llamada este disponible para el procedimiento remoto, el puntero o la referencia a la variable global estando fuera de uso. En

la práctica, La mayoría de RPCs restringen el uso de variables globales, para evitar la necesidad enviar el contexto del proceso completo a través de la red. En lugar de enviar un puntero, la mayoría de RPCs no referenciará el puntero y enviara el objeto al cual apunta.

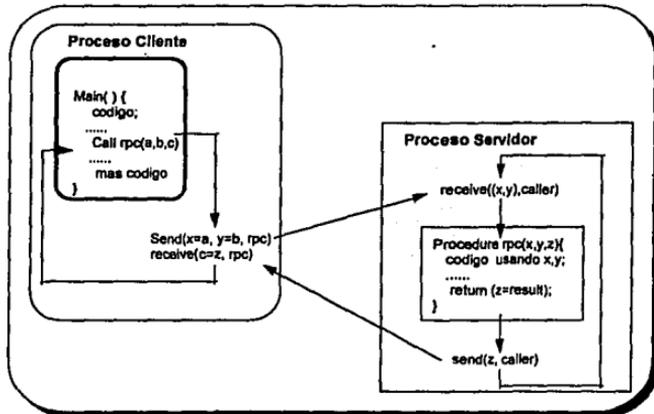


Figura. 3.17 Llamada a un procedimiento remoto

Los RPCs son inherentemente sincronizados, ya que raramente tiene sentido llamar un procedimiento sin resultados de regreso. El par emisor/receptor que es ejecutado por el cliente debe bloquear las operaciones para asegurar que el resultado este siempre regresando de el RPC. Esto es fácilmente logrado, ya que emisor y el receptor están fuera del ámbito de el programa de aplicación. Los programadores de aplicaciones simplemente enlazan su programa a la librería del RPC en el nodo que llama y su procedimiento llamado a la librería del RPC en el nodo destino.

Esto es mostrado en mayor detalle en la figura 3.18. La librería del RPC sobre la máquina cliente proporciona un módulo de código para el procedimiento remoto. Este módulo de código contiene un par emisor/receptor y es enlazado al código de la aplicación para formar el proceso del cliente. La librería del RPC en el servidor proporciona una infraestructura para el procedimiento remoto. Esta infraestructura contiene un par emisor/receptor con el código necesario para invocar al procedimiento remoto. Cuando se enlaza con el código de la aplicación constituye el proceso del servidor. Para el programador de aplicaciones es casi tan trivial como llamar a procedimientos locales, ya que el protocolo es transparente.

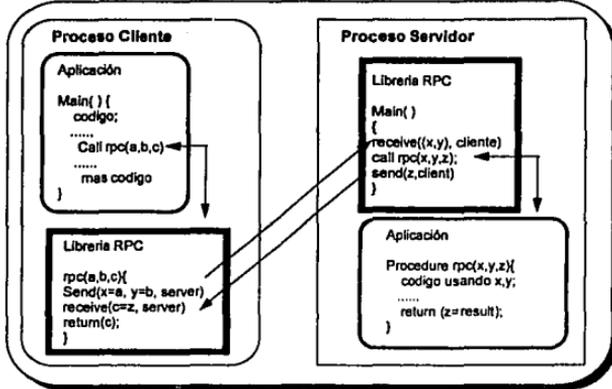


Figura 3.18 Implementación de la librería RPC

3.4.1. PRINCIPALES COMPONENTES DE UN RPC

Una buena ilustración de un RPC es el seleccionado por la Fundación de Software Abierto (OSF, Open Software Foundation) basado en el RPC de Hewlett-Packard.

1. *Librería de tiempo de corrida (run-time library)* implementa el protocolo de RPC. A estas funciones corresponde en gran parte las facilidades de intercambio de datos de extremo a extremo en estructuras previas. La librería de tiempo de corrida del RPC tiene un conjunto de interfaces (run-time API) para usarse por el módulo de código para la llamada, este módulo es utilizado para buscar información en un llamado al servidor o un directorio de sistema, establecer conexión a el servidor, enviar llamadas de solicitudes y respuestas, etc. Una responsabilidad clave de la librería de tiempo de corrida es la de mantener el control de las condiciones de estado en nombre del cliente y servidor durante una conexión. Finalmente, la librería del tiempo de corrida construye los encabezados apropiado para los paquetes del RPC, como es requerido por el suministrador del servicio de transporte e invoca al suministrador del servicio de transporte en nombre del cliente y el servidor.

El lenguaje para Definición de Interface en la Red (Network Interface Definition Language NIDL)¹, para definir la interface de RPC y producir el módulo de código para la llamada. Es un lenguaje estrictamente declarativo. Contiene solo construcciones para definir constantes, tipos y operaciones de una interface; no tiene construcciones ejecutables (el NIDL esta considerado para ser una función de la capa 7 de OSI). El

¹ T.H.Dineen, the network computing architecture and system, february 1988.

NIDL fuente esta compuesto con declaraciones de un lenguaje de alto nivel, aumentado con definiciones de la interface e indicaciones para el uso de cada parámetro. Otra información en el NIDL donde es hecha la conversión de datos, tipo de parámetros ordenados para ser usados, soporte para la verificación del servidor y si la librería de tiempo de corrida soportará protocolos de recuperación de dos fases.

2. *El compilador NIDL* (algunas veces conocido como el procesador NIDL o generador del módulo de código para la llamada). Checa la sintaxis, traduce las definiciones NIDL a declaraciones en lenguajes como C y genera el módulo de código para la llamada para usarse en el cliente y en el servidor.
3. *El módulo de código para la llamada (stub)* en acuerdo con la librería del tiempo de corrida del RPC, conecta los programas de llamada al ambiente de tiempo de corrida (fig. 3.19). EL módulo de código para la llamada del cliente primero emitirá un comando `RpcBindToInterface` , incluyendo la sintaxis de transferencia propuesta y el esquema de transporte, para intentar la vinculación. La información retomada puede indicar una opción una sintaxis de transferencia y/o direcciones de transporte para ser usadas. El módulo de código para la llamada buscara seleccionar una de cada una y continuar con el actual proceso de llamada.

Un comando `RpcSend` manda la transmisión de una solicitud de llamada a un servidor. Un comando `RpcReceive` obtendrá los resultados del servidor. Un comando `RpcUnbind` terminara la vinculación.

4. *La Representación de Datos en Red* (Network Data Representation NDR) es un protocolo que define como los valores estructurados que son pasados en el RPC están codificados para la transmisión de la red. El propósito de NDR es habilitar máquinas con diferente representación local de datos para comunicar un tipo de valores de una a otra.

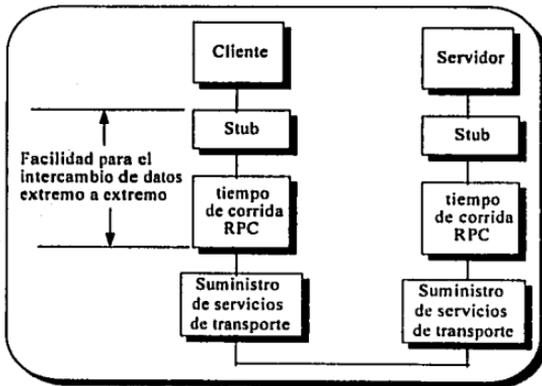


Figura 3.19 Estructura RPC

El protocolo NDR especifica un conjunto de tipo de datos que pueden ser usados para especificar un conjunto de tipo de valores, además define como representar formatos escalares. NDR representa una secuencia de valores con un formato de etiqueta más una cadena de bits.

3.4.2. MODELOS DE INTERACCION

La mayoría de los protocolos de red requieren que la librería del RPC en el servidor este corriendo antes de que el primer RPC use sus servicios. Protocolos más avanzados son capaces de iniciar un proceso con el servidor aunque la librería no este ya ejecutándose, esto es hecho considerando el nombre del servidor en un archivo de configuración basado en el tipo de mensaje recibido. Si la librería del servidor no esta corriendo, el archivo de configuración indica como arrancar.

Una vez arrancado, este bloque en el servidor con una recepción, espera solicitudes del cliente, cuando una solicitud llega, el servidor responde, llamando el procedimiento de la aplicación, que regresa sus resultados y cierra el ciclo con un receive para la próxima solicitud.

Otra solución es suministrar un procesador maestro que contiene la operación receive pero invoca al procedimiento remoto como un procesador esclavo. El maestro está entonces inmediatamente disponible para servir a la próxima llamada RPC. Esto es ilustrado en la figura 3.20

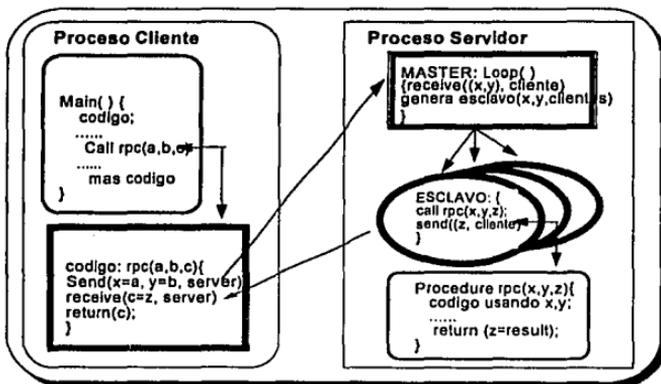


Figura 3.20 Procesos Maestro/esclavo en el servidor

Hasta se ha descrito un estricto modelo RPC, en el cual el servidor esta ocioso de que entren solicitudes de los clientes. Con el modelo de reunión (rendezvous) ambas partes del proceso se ejecutan independientemente pero acuerdan reunirse periódicamente para intercambiar mensajes. En una reunión cliente/servidor la solicitudes del cliente se ejecutan en un punto de entrada específico del servidor. Cuando el servidor esta listo para ejecutar este punto de entrada, emite un comando `accept(punto-entrada,mensaje)`. El

accept es lo mismo que **receive** (cualquier mensaje) excepto que el primero sólo acepta mensajes para un punto de entrada específico.

Las diferencias entre el modelo de reunión y el modelo estricto RPC es que en la reunión, el servidor ejecuta el **accept** sólo cuando se encuentra listo, no cuando el cliente demanda el servicio. Además, el servidor puede especificar que tipo de llamada esta ordenando el próximo servicio. En el modelo estricto RPC, el servidor debe invocar cualquier procedimiento que el cliente pida y por lo tanto debe ser capaz de ejecutarlos de forma independiente, por último desde la perspectiva del tiempo de corrida, el modelo estricto RPC implica múltiples hilos de ejecución a través del servidor. Cada solicitud enviada al servidor requiere alojamiento y desalojamiento dinámico de los recursos del servidor, el servidor debe crear un contexto separado (ej. pila de tiempo de corrida) en el cual ejecutar cada solicitud. Una reunión se ejecuta dentro del contexto de un servidor de procesos ya corriendo y es inherentemente armada usando hilos individualmente.

Otra desviación del modelo estricto RPC involucra el uso de llamadas devueltas (callbacks). Usando llamadas devueltas, el cliente proporciona el nombre de un procedimiento que el servidor puede llamar si ocurre una excepción durante el procesamiento del RPC. Las llamadas devueltas son frecuentemente usadas para dar notificaciones al cliente de eventos significantes ocurriendo en el servidor.

3.4.3. TIPOS DE FALLAS

Hay cuatro tipos de fallas para un RPC:

1. Caída del cliente después de llamar un procedimiento remoto.
2. El mensaje llamando al procedimiento remoto puede perderse.
3. Caída del servidor mientras ejecuta el procedimiento remoto.
4. El mensaje regresando los resultados puede perderse.

Si el cliente se cae después de llamar un procedimiento remoto, el servidor hará el trabajo para nada. El mensaje al regresar los resultados deberá ser descartado por el subsistema de comunicaciones del cliente, porque sería un proceso inexistente. Aun si el cliente reinicia, el protocolo debe distinguir entre un proceso actual y un proceso previo y descartar el mensaje anterior.

Si los mensajes llamando o regresando desde el procedimiento remoto están perdidos, el cliente puede potencialmente esperar para siempre. La mayoría de protocolos RPC dependen de reconocimientos (ACK's) para asegurar que estos mensajes estén correctamente recibidos. Ya que el servidor no tiene forma de saber si su ACK fue recibido, debe estar atento y ejecutar la primera llamada. Por lo tanto, el servidor debe tener algún registro de llamadas previas para no procesar la misma llamada dos veces. Esto es conocido como *no más de una vez*. La semántica de un RPC es generalmente implementada agregando un número secuencial a cada solicitud. Si una llamada es reenviada, su número secuencial deberá ser el mismo que antes.

Si el cliente no recibe el ACK como respuesta del mensaje, el servidor puede esperar cierto tiempo y usar su memoria de llamadas procesadas para reenviar la respuesta al cliente. Si el ACK de respuesta al mensaje esta

perdido, el cliente recibirá dos respuestas a la misma llamada y deberá saber como descartar la segunda. La figura 3.21 ilustra como un protocolo de tres paquetes (solicitud, respuesta y reconocimiento) puede ser usado para procesamiento de fallas.

Si el servidor cae, sus clientes esperaran cierto tiempo y reenviaran sus llamadas. Si el servidor esta fuera por mucho tiempo, los intentos repetidos de estos clientes pueden causar una congestión en la red. Después de algún número de reintentos, los clientes deben desistir dependiendo si las llamadas devueltas o el usuario saben cuando reiniciará el servidor.

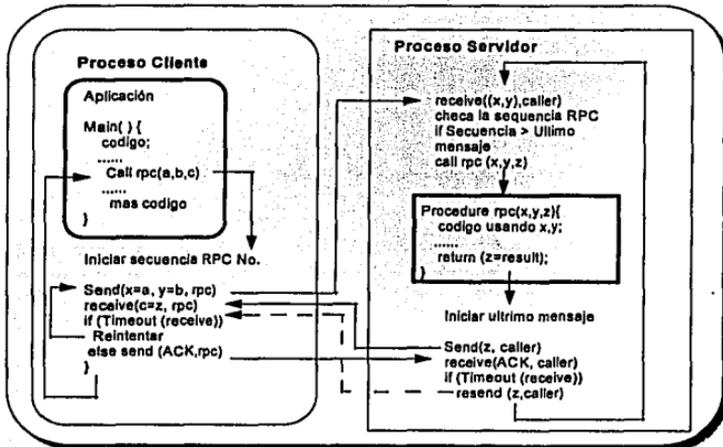


Figura 3.21 Ejemplo de protocolo de tres paquetes

En la práctica, un servicio de RPC orientado a conexión garantiza una entrega confiable de mensajes en el orden en que fueron enviados. Un servicio de RPC no orientado a conexión es usualmente construido en base a un servicio de transporte con datagramas y no garantiza que todos los mensajes enviados sean recibidos.

3.4.4. SEGURIDAD

Para ser completamente seguro, un protocolo RPC debe proporcionar una forma de:

- Identificar el emisor del mensaje
- Verificar la identidad del emisor
- Controlar como es ruteado el mensaje
- Detectar alguna interferencia con el mensaje y encriptar el mensaje

No todos los RPC's proporcionan los cuatro aspectos, de hecho, el RPC no puede siempre proporcionar los cuatro aspectos.

CAPITULO IV

SISTEMAS

OPERATIVOS

DE RED

CAPITULO IV: SISTEMAS OPERATIVOS DE RED

INTRODUCCION

El programa fundamental de todos los programas de sistemas es el SISTEMA OPERATIVO, que controla todo los recursos de la computadora y proporciona las bases sobre la cual pueden escribirse los programas de aplicación.

Uno de los obstáculos al utilizar las computadoras era el manejo del Hardware, de aquí surge la necesidad de determinar la forma de evitar la complejidad del manejo del Hardware, esto se ha logrado gracias a que se ha colocado un nivel de software por arriba del hardware con el fin de controlar todas las partes del sistema y presentar al usuario una interfaz o máquina virtual que facilite la comprensión del programa. Este nivel de Software es el Sistema Operativo. En la figura 4.1 se muestra el sistema de Software en varias capas:

Sistema de Nómina	Programas del Usuario	Juegos	Programa de Aplicación
Compiladores	Editores	Interprete de Comandos	Programa de Sistema
Sistema Operativo			Hardware
Lenguaje de Máquina			
Microprogramación			
Dispositivos Físicos			

Figura 4.1 Sistema de Software

La capa de Hardware se encuentra dividida en tres categorías:

- *Dispositivos Físicos*, conformados por chips, cables, fuentes de poder, etc.
- *Microprogramación*, es el software que controla en forma directa los dispositivos y proporciona, una interfaz más limpia con la siguiente capa.
- *Lenguaje de Máquina*, es el conjunto de instrucciones que interpreta el miniprograma.

La capa de Programas de Sistemas se encuentra dividido en dos categorías:

- *Sistemas Operativos*, oculta las capas de Hardware proporcionando al usuario un conjunto más conveniente de instrucciones con el cual trabajar.
- *Los compiladores, editores e interprete de instrucciones*, son utilerías que se ejecutan en modo usuario.

Los Programas de Aplicación, son escritos por el usuario para resolver sus necesidades particulares.

Los sistemas operativos se han venido desarrollando de alguna manera en forma paralela con el desarrollo de las máquinas. Uno de los desarrollos más importantes desde la década de los 80' ha sido el crecimiento de las redes de computadoras personales con sistemas operativos de red y sistemas operativos distribuidos. En un sistema operativo de red, los usuarios están consientes de la existencia de varias

computadoras y pueden conectarse con máquinas remotas y copiar archivos de una máquina a otra. Cada máquina tiene su propio sistema operativo local y su propio usuario (o grupo de usuarios).

Un sistema operativo distribuido es aquel que aparece ante sus usuarios como un sistema tradicional de un sólo procesador. En un sistema operativo distribuido verdadero, los usuarios no deben de ser conscientes del lugar donde su programa se ejecute o del lugar donde se encuentran los archivos; esto debe de ser manejado en forma automática y eficaz por el sistema operativo.

Los sistemas operativos de red no tienen diferencias fundamentales con los sistemas operativos de un sólo procesador, es obvio que necesitan un controlador de interfaz de la red y algo de software de bajo nivel para dirigirlo, al igual que programas que permitan la conexión y el acceso a un archivo remoto, pero estas características adicionales no modifican la estructura del sistema operativo.

Los verdaderos sistemas operativos distribuidos requieren añadir un poco más de código a un sistema operativo de un único procesador. Estos permiten a menudo que un programa se ejecute mediante varios procesadores a la vez, por lo que necesitan algoritmos de asignación de tiempo más complejos, con el fin de optimizar la magnitud de paralelismo logrado.

La interfaz entre el sistema operativo y los programas de usuario se define como un conjunto de "instrucciones ampliadas" que proporciona el sistema operativo. Estas instrucciones ampliadas se conocen como llamadas al sistema. Estas crean, eliminan y utilizan varios objetos del software, controlados por el sistema operativo. Las más importantes son los procesos y archivos. Los cuales serán discutidos más ampliamente en este capítulo enfocando las características que los convierten en sistemas distribuidos.

4.1 PARTES PRINCIPALES DE UN SISTEMA OPERATIVO:

KERNEL O NUCLEO: Es la parte más interna de un sistema operativo, teniendo tres tareas muy importantes:

- Ver a qué proceso le asignará el procesador. Usando el despachador (dispatcher).
- El manejo de interrupciones.
- La comunicación entre procesos (IPCS): Es una forma general que brindan los sistemas operativos para sincronizar actividades entre procesos. La idea general de esta comunicación es establecer un sistema de transmisión de mensajes entre los procesos, funcionando, básicamente con las operaciones de *enviar(mensajes)* y *recibir(mensajes)*.

MANEJADOR DE MEMORIA: Proceso encargado de asignar espacio en memoria para los demás procesos que se encuentran corriendo y de liberar el espacio para aquellos que han sido terminados. Para ello utiliza alguno de los métodos que se describen posteriormente.

MANEJO DEL PROCESADOR: Este programa del sistema (a veces conocido como despachador de alto nivel o scheduler) se encarga de determinar el orden óptimo de atención a los diversos procesos que están compitiendo por ganar la atención del procesador central. En la sección de procesos se ampliará la información de este punto.

MANEJADOR DE ENTRADAS/SALIDAS: Este se involucra cuando un proceso necesita un dato de entrada o requiere enviar a algún dispositivo. En este caso el propio proceso genera una interrupción y el sistema operativo llama a este manejador. De manera típica cuando este subsistema ha manejado apropiadamente el dato, genera una instrucción de retorno de interrupción y el programa llamador continúa. Estos subsistemas pueden trabajar con el concepto de poner en cola de espera los requerimientos (queues).

MANEJO DE INFORMACION (File System): Este sistema se encarga de conocer y manejar a los dispositivos de almacenamiento, y por otro lado de recibir y ejecutar las instrucciones que envían los demás procesos para grabar o leer datos.

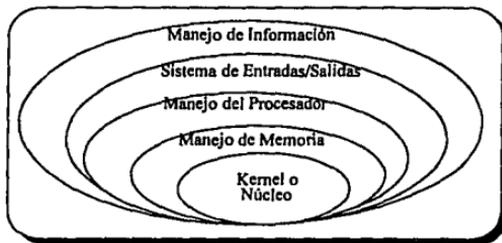


Figura 4.2 Partes del Sistema Operativo

4.2 MODELOS DE SISTEMAS OPERATIVOS

4.2.1. MODELO MONOLITICO

Se basa en un gran Kernel lleno de procedimientos que se comunican entre sí realizando solicitudes uno a otro, este modelo tiene la desventaja de que si existe un procedimiento que cause un error, el resto del sistema se verá afectado pues el Kernel se comporta como una unidad.

El sistema se describe como una colección de procedimientos, cada uno de los cuales puede llamar a los demás cada vez que así lo quiera. Cuando se usa esta técnica, cada procedimiento del sistema tiene una interfaz bien definida en términos de parámetros y resultados, cada uno de ellos es libre de llamar a cualquier otro, si este último proporciona cierto cálculo útil para el primero.

En los sistemas monolíticos los servicios (llamadas al sistema) que proporciona el sistema operativo se solicita colocando los parámetros en lugares bien definidos, como en registros o en la pila, para después ejecutar una instrucción especial de trampa de nombre llamada al núcleo o llamada al supervisor.

Esta instrucción cambia la máquina del modo usuario al modo núcleo (también conocido como modo supervisor) y transfiere el control al sistema operativo, lo que se muestra en el evento (1) de la figura 4.3. (La mayoría de los CPU tienen 2 modos en el modo núcleo para el sistema operativo, en el que permite todas las instrucciones; el modo usuario, para los programas del usuario, en donde no se permiten E/S y ciertas instrucciones más).

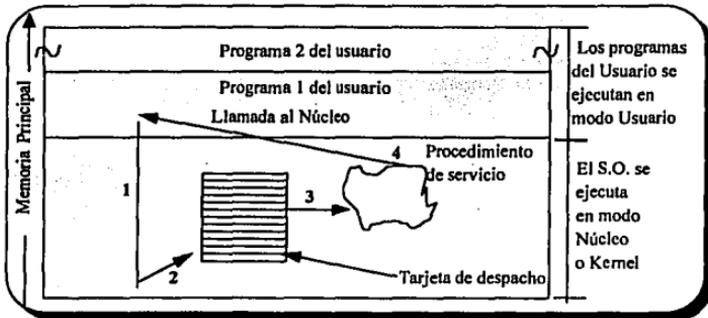


Figura 4.3 Llamada al sistema:

En la figura 4.3 se ilustran cuatro pasos para hacer una llamada al sistema y estos son:

1. El programa del usuario es atraído hacia el núcleo, 2. El SO determina el número de servicio solicitado.
3. El SO localiza y llama al procedimiento correspondiente al servicio. 4. El control regresa al programa del usuario.

Esta organización sugiere una estructura básica del Sistema Operativo:

1. Un programa principal que llama al procedimiento del servicio solicitado.
2. Un conjunto de procedimientos de servicio que lleva a cabo las llamadas al sistema.
3. Un conjunto de procedimientos utilitarios que ayudan al procedimiento de servicio.

En este modelo para cada llamada al sistema existe un procedimiento de servicio que se encarga de él. Los procedimientos utilitarios hacen cosas necesarias para varios procedimientos de servicio. La división de procedimientos se realiza en 3 capas:

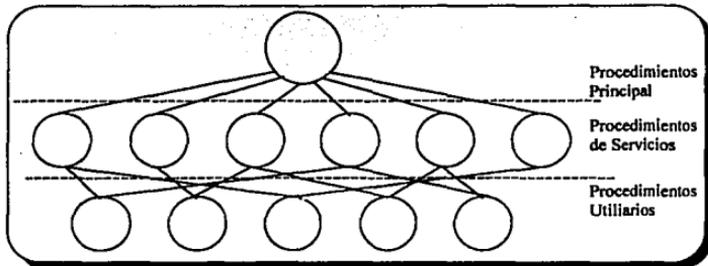


Figura 4.4 Un modelo de Estructura Simple para un Sistema Monolítico

4.1.2. SISTEMAS CON CAPAS

Este sistema consiste en organizar el sistema operativo como una jerarquía de capas, cada una construida sobre la inmediata inferior. Se considera como un sistema sencillo de procesamiento por lotes. El sistema consiste en 6 capas como se muestra en la figura 4.5. La capa 0 trabaja con la asignación del procesador y alterna entre los procesos cuando ocurren las interrupciones o expiran los cronómetros. Sobre la capa 0, el sistema consta de procesos secuenciales, es decir, en esta capa proporciona la multiprogramación básica de la CPU.

- La capa 1 realiza la administración de la memoria.
- La capa 2 se encarga de la comunicación entre cada proceso y la consola del operador.
- La capa 3 controla los dispositivos de E/S y guarda en almacenes (buffers) los flujos de información entre ellos.
- La capa 4 se encuentran los programas del usuario
- La capa 5 se encuentran los procesos del operador.

5	El Operador
4	Programas del usuario
3	Control de Entrada/Salida
2	Comunicación Operador-Proceso
1	Administración de memoria y disco
0	Asignación del procesador y multitarea

Figura. 4.5 Sistema por Capas

4.1.3. MODELO CLIENTE/SERVIDOR

Una tendencia de los sistemas operativos modernos es la de explotar la idea de mover el código a capas superiores y eliminar la mayor parte posible del sistema operativo para mantener un núcleo mínimo. El punto de vista usual es el de implantar la mayoría de las funciones del sistema operativo en los procesos del usuario. Para

solicitar un servicio, como la lectura de un bloque de un archivo, un proceso del usuario (denominado Proceso Cliente) envía la solicitud a un proceso servidor que realiza entonces el trabajo y regresa la respuesta.

En este modelo lo único que hace el núcleo es controlar la comunicación entre los clientes y los servidores. Al separar al sistema operativo en partes, cada una de ellas controla una faceta del sistema, como el servicio a archivos, servicios de proceso, servicios a terminales o servicios a la memoria, cada parte es pequeña y controlable. Como todos los servicios se ejecutan como procesos en modo usuario y no en modo núcleo, no tiene acceso directo al hardware. En consecuencia si hay un error en el servidor de archivos, este podría fallar pero no afecta en general a toda la máquina.

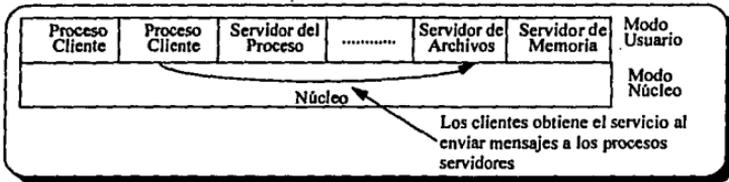


Figura 4.7 El modelo Cliente/Servidor

Otra de las ventajas del modelo Cliente/Servidor es la capacidad de adaptación para su uso en los sistemas distribuidos. Si un cliente se comunica con el servidor mediante mensajes, el cliente no necesita saber si el mensaje se maneja en forma local, en su máquina o si se envía por medio de una red a un servidor en una máquina remota. En lo que respecta al cliente, lo mismo ocurre en ambos casos; se envió una solicitud y se recibió un respuesta.

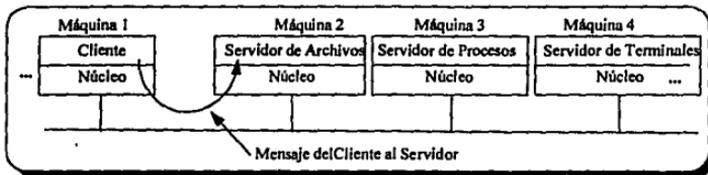


Figura 4.8 El modelo C/S en un Sistema Distribuido

La idea anterior de un núcleo que sólo controla el transporte de mensajes de clientes a servidores y viceversa no es totalmente real. Algunas funciones del sistema operativo (como el cargador de comandos en los registros físicos de los dispositivos de E/S) son difíciles, si es no que imposibles de realizar a partir de programas del usuario. Existen dos formas de resolver dicho problema. Uno es la de hacer que algunos procesos de servidores críticos (por ejemplo, las directivas de dispositivos de E/S) se ejecutan en realidad en modo núcleo, con acceso total al hardware, pero de forma que se comuniquen con los demás procesos mediante el mecanismo normal de mensajes.

La otra forma es la de construir una cantidad mínima de mecanismos dentro del núcleo, pero manteniendo las decisiones de política relativas a los usuarios dentro de los espacios del usuario. Por ejemplo el núcleo podría reconocer que cierto mensaje enviado a una dirección especial indica que se tome el contenido de ese mensaje y se cargue en los registros de dispositivos de E/S de algún disco, para iniciar la lectura del disco.

4.3 PROCESOS DISTRIBUIDOS

El concepto central de cualquier sistema operativo es el *proceso*. Un proceso es un programa en ejecución.

Todas las computadoras modernas hacen varias cosas al mismo tiempo. En un sistema multiprogramación, la CPU también alterna de programa en programa, ejecutando cada uno de ellos por decenas o cientos de milisegundos. Aunque, en sentido estricto, la CPU ejecuta en cierto instante un sólo programa, durante un segundo puede trabajar con varios de ellos, lo que da una apariencia de paralelismo. A veces llamado *pseudoparalelismo*.

Aunque cada proceso es una entidad independiente, con su propio contador de programa y estado interno, es frecuente que los procesos deban interactuar con otros. Esto nos lleva a lo que se conoce como estado de procesos. en la figura 4.9 vemos un diagrama de estados, el cuál muestra las tres funciones que puede tener un proceso:

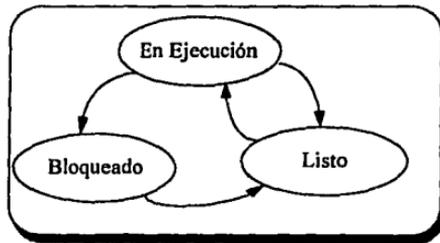


Figura 4.9 Transiciones entre estados

1. En ejecución (utiliza la CPU en un instante dado) .
2. Listo (ejecutable, se detienen en forma temporal para que se ejecute otro proceso) .
3. Bloqueado (no se puede ejecutar debido a la ocurrencia de algún evento externo) .

Desde el punto de vista lógico los dos primeros estados son similares. En ambos casos el proceso desea ejecutarse, sólo que en el segundo caso, no existe CPU disponible para él. El tercer estado es distinto de los otros puesto que el proceso no se puede ejecutar, incluso aunque la CPU no tenga labores que realizar.

4.3.1. COMUNICACION ENTRE PROCESOS (IPC)

Los procesos requieren con frecuencia la comunicación entre ellos. Por ejemplo un entubamiento, la salida del primer proceso debe transferirse al segundo proceso y así sucesivamente. De la misma manera existe la necesidad de comunicación entre procesos; de preferencia, en una forma estructurada sin utilizar interrupciones.

Se describirán dos métodos de comunicación entre procesos que aunque por sí mismos no se pueden implantar en sistemas operativos distribuido. Se logra su implantación usando la transferencia de mensajes (propia de sistemas distribuidos).

SEMAFOROS

Existen dos primitivas de comunicación entre procesos que son: dormir (sleep) y despertar (wakeup). sleep es una llamada al sistema que provoca el bloqueo de quién hizo la llamada, es decir, que sea suspendido hasta que otro proceso lo despierte, la llamada wakeup tiene un parámetro, el proceso por despertar.

En 1965 se sugirió el uso de una variable entera, llamada semáforo, para contar el número de despertares almacenados para sus uso posterior. Un semáforo puede tener un valor 0, lo que indica que no existen despertares almacenados; o bien algún valor positivo si están pendientes uno o más despertares.

Posteriormente se propusieron dos operaciones, down y up (generalizaciones de sleep y wakeup, respectivamente). La operación down verifica si el valor de un semáforo es mayor que 0. En este caso, decrementa el valor (es decir, utiliza un despertar almacenado) y continúa. Si el valor es 0, el proceso se va a dormir. La verificación y modificación del valor, así como la posibilidad de irse a dormir se realiza en conjunto, como una sola e indivisible acción atómica. Se garantiza que al iniciar una operación con un semáforo, ningún otro proceso puede tener acceso al semáforo hasta que la operación termine o se bloquee. Esta atomicidad es absolutamente esencial para resolver los problemas de sincronización y evitar condiciones de competencia entre procesos.

La operación up incrementa el valor del semáforo correspondiente. Si uno o más procesos dormían en ese semáforo y no podían completar una operación down anterior, el sistema elige algunos de ellos (por ejemplo, en forma aleatoria) y se le permite terminar down. Así, después de un up en un semáforo con procesos durmiendo, el semáforo seguirá con valor 0, pero habrá un número menor de procesos durmiendo. La operación de incremento del semáforo y despertar de un proceso también es indivisible.

MONITORES

Se debe tener cuidado al utilizar semáforos. Un pequeño error y todo se vendría abajo.

Para facilitar la escritura de programas correctos, se propuso una primitiva de sincronización de alto nivel, llamada monitor. Un monitor es una colección de procedimientos, variables y estructuras de datos que se agrupan en cierto tipo particular de módulo o paquete. Los procesos pueden llamar a los procedimientos de un

monitor cuando lo deseen, pero no tiene acceso directo a las estructuras de datos ínternas del monitor desde los procedimientos declarados fuera de él.

Los monitores tienen una propiedad importante que los hace útiles para conseguir la exclusión mutua: sólo uno de los procesos puede estar activo en un monitor en cada momento. Los monitores son construcciones del lenguaje de programación, por lo que el compilador sabe que son especiales y puede controlar las llamadas a los procedimientos del monitor en forma distinta a las llamadas de los demás procedimientos. Por lo general, cuando un proceso llama a un procedimiento monitor, las primeras instrucciones de éste verificarán si hay otro proceso activo dentro del monitor. En caso afirmativo, el proceso que hace la llamada será suspendido hasta que el otro proceso salga del monitor. Si no hay otro proceso que esté utilizando el monitor, el que hace la llamada podrá entrar.

USO DE LOS MENSAJES PARA LA IMPLANTACION DE SEMAFOROS Y MONITORES

Si se dispone de un sistema de mensajes, se pueden implantar los semáforos y los monitores mediante un pequeño truco. Este consiste en introducir un nuevo proceso, el proceso de sincronización. Se verá primero la forma en que se puede utilizar este proceso para implantar los semáforos. El proceso de sincronización mantiene, para cada semáforo, un contador y una lista ligada en procesos de espera. Para llevar a cabo un up o un down, un proceso llama al correspondiente procedimiento de biblioteca, up o down, el cual envía al proceso de sincronización un mensaje que incluye la operación deseada y el semáforo por utilizar. El procedimiento de biblioteca recibe entonces un RECEIVE para obtener la réplica del proceso de sincronización.

Cuando llega el mensaje, el proceso de sincronización verifica el contador para ver si la operación pedida se puede llevar a cabo. Los up siempre se pueden llevar a cabo, pero los down se bloquean si el valor del semáforo es 0. Si la operación es permitida, el proceso de sincronización envía de regreso un mensaje vacío, con lo que elimina el bloqueo del proceso que hizo la llamada. Si, por el contrario, la operación es un down y el semáforo es 0, el proceso de sincronización forma el proceso que hizo la llamada en la cola y no envía una réplica. El resultado es el bloqueo del proceso que realiza el down, lo cual es correcto. Más tarde, si se realiza un up, el proceso de sincronización elige uno de los procesos bloqueados en el semáforo, ya sea por orden de llegada (el primero en llegar es el primero al que se da servicio) o de prioridad o cualquier otro orden y les envía una réplica. Las condiciones de competencia se evitan en este caso debido a que el proceso de sincronización sólo trabaja con una solicitud a la vez.

Los monitores se pueden implantar por medio de mensajes con el mismo truco. Una forma de lograr esto es haciendo que el compilador implante los procedimientos del monitor mediante llamadas a los procedimientos de biblioteca up y down para asociarle un semáforo adicional, llamado mutex y semáforos para cada proceso,

como se describió al principio de esta sección. Estos procedimientos serán implantados entonces mediante el envío de mensajes al proceso de sincronización. También son posibles otras implantaciones.

4.3.2. HILOS (THREADS)

En la mayoría de los sistemas operativos tradicionales, cada proceso tiene un espacio de direcciones y un único hilo de control. Sin embargo, con frecuencia existen situaciones en donde se desea tener varios hilos de control que compartan un único espacio de direcciones, pero que se ejecutan de manera quasi-paralela, como si fuesen de hecho procesos independientes (excepto por el espacio de direcciones compartido).

En la figura 4.10 (a) se observa una máquina con tres procesos. Cada uno de ellos tiene su propio contador del programa, su propia pila, su propio conjunto de registros y su propio espacio de direcciones. Los procesos no tienen nada que ver entre sí, excepto que podrían comunicarse mediante las primitivas de comunicación entre procesos del sistema, como los semáforos, monitores o mensajes. En la figura 4.10 (b) se observa otra máquina, con un proceso, sólo que ahora este proceso tiene varios hilos de control, los cuales, por lo general, se llaman simplemente hilos o a veces procesos ligeros. En muchos sentidos, los hilos son como miniprocesos. Cada hilo se ejecuta en forma estrictamente secuencial y tiene su propio contador de programa y una pila para llevar un registro de su posición. Los hilos comparten el CPU, de la misma forma que lo hacen los procesos: primero, se ejecuta un hilo y después otro (tiempo compartido). Sólo en un multiprocesador se pueden ejecutar realmente en paralelo. Los hilos pueden crear hilos hijos y se pueden bloquear en espera de que se terminen llamadas al sistema, al igual que los procesos regulares. Mientras un hilo está bloqueado, se puede ejecutar otro hilo del mismo proceso, exactamente en la misma forma en que, cuando se bloquea un proceso, se puede ejecutar en la misma máquina otro proceso.

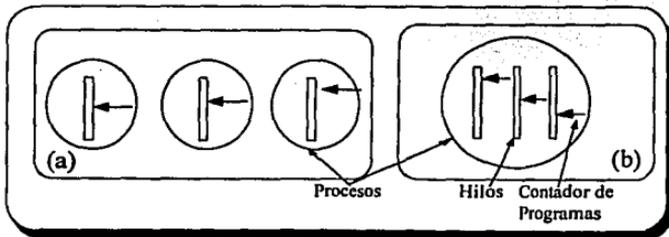


Figura 4.10 (a) Tres procesos con un hilo c/u (b) Un proceso con tres hilos

Como los procesos tradicionales (es decir, procesos con un único hilo) los hilos pueden tener uno de los siguientes estados: en ejecución, bloqueado, listo o terminado.

USO DE HILOS.

Los hilos se inventaron para permitir la combinación del paralelismo con la ejecución secuencial y el bloqueo de las llamadas al sistema. En la figura 4.11 (a) se muestra una posible organización, en la que un hilo, el servidor, lee las solicitudes de trabajo en el buzón del sistema. Después de examinar la solicitud elige a un hilo, trabajador inactivo (es decir, bloqueado) y le envía la solicitud, lo cual se realiza con frecuencia al escribir un apuntador al mensaje en una palabra especial asociada a cada hilo. El servidor despierta entonces al trabajador dormido (por ejemplo, lleva a cabo un up en el semáforo en donde duerme).

Cuando el trabajador despierta, verifica si puede satisfacer la solicitud por medio del bloque caché compartido, al que tienen acceso todos los hilos. Si no, envía un mensaje al disco para obtener el bloque necesario (si se trata de un READ) y se duerme en espera de la conclusión de la operación del disco. Se llama entonces al planificador y se inicializa otro hilo, que tal vez sea el servidor, para pedir más trabajo; o bien a otro trabajador listo para realizar un trabajo.

La estructura del servidor en la figura 4.11 (a) no es la única manera de organizar un proceso de muchos hilos. El modelo de equipo de la figura 4.11 (b) es también una posibilidad. Aquí todos los hilos son iguales y cada uno obtiene y procesa sus propias solicitudes. No hay servidor. A veces llega trabajo que un hilo no puede manejar, en particular si cada hilo se especializa en manejar cierto tipo de trabajo. En este caso, se puede utilizar una cola de trabajo, la cual contiene todos los trabajos pendientes. Con este tipo de organización, un hilo debe verificar primero la cola de trabajo antes de buscar en el buzón del sistema.

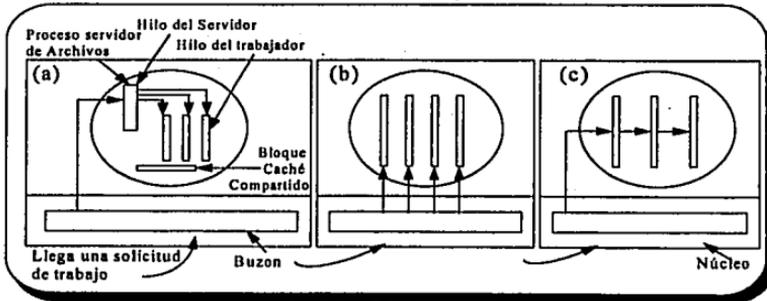


Figura 4.11 Tres organizaciones de hilos en un proceso a) Modelo del servidor/trabajo. b) Modelo de equipo. c) Modelo de entubamiento

Los hilos se pueden organizar también mediante el modelo de entubamiento de la figura 4.11 (c). En este modelo, el primer hilo genera ciertos datos y los transfiere al siguiente para su procesamiento. Los datos pasan de hilo en hilo y en cada etapa se lleva a cabo cierto procesamiento. Los entubamientos se utilizan ampliamente

en muchas áreas de los sistemas de cómputo, desde la estructura interna de los CPU RISC hasta las líneas de comandos de UNIX.

IMPLANTACION DE UN PAQUETE DE HILOS.

Existen dos vías para implantar un paquete de hilos: en el espacio del usuario y en el espacio del núcleo.

El primer método consiste en colocar todo el paquete de hilos en el espacio del usuario. El núcleo no sabe de su existencia. En lo que respecta a éste, maneja procesos ordinarios con un único hilo.

Todas estas implantaciones tienen la misma estructura general, la cual se muestra en la figura 4.12 (a). Los hilos se ejecutan en la parte superior de un sistema de tiempo de ejecución, el cual es una colección de procedimientos que manejan los hilos. Cuando un hilo ejecuta una llamada al sistema, se duerme, desarrolla una operación en un semáforo.

Consideremos ahora que el núcleo sabe de la existencia de los hilos y debe manejarlos. No se necesita un sistema de tiempo de ejecución, como se muestra en la figura 12-8 (b). La información es la misma que en el caso de los hilos a nivel usuario, sólo que ahora se encuentra en el espacio del núcleo y no en el espacio de usuario (dentro del sistema de tiempo de ejecución).

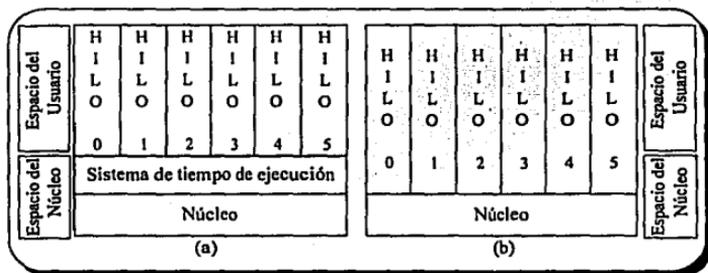


Figura 4.12 (a) Un paquete de hilos a nivel a usuario. (b) Paquetes de hilos manejado por el núcleo

4.4 ADMINISTRACION DE LA MEMORIA

Existen varias maneras de manejar la memoria en un sistema de cómputo. Aquí explicaremos los esquemas que permiten la multiprogramación.

4.4.1. MANEJO DE MEMORIA POR PARTICIONES (FRAGMENTACION)

Es el primer esquema para permitir la multiprogramación (activación de varios procesos simultáneamente) y consiste en subdividir la memoria en varias secciones fijas y asignar cada una de ellas a un usuario o proceso activo. El principal problema por resolver es asegurar que ningún usuario intervenga en el área de memoria asignada a otro. Desde este punto de vista, el manejo de memoria consiste en controlar cuáles

particiones están asignadas a cuáles procesos, para poder liberar particiones cuando los procesos residentes en ellas terminen o cambien, y poder ofrecer particiones libres a procesos que soliciten atención por parte del sistema de cómputo.

La ventaja fundamental de este modelo es que permite la multiprogramación, y su principal desventaja consiste en que deja lugares libres en la memoria que, como son de tamaño fijo, no pueden ser utilizados más que por procesos de longitud menor o igual a la de la partición en cuestión.

4.4.2. PARTICIONES RELOCALIZABLES

La desventaja del esquema anterior llamado fragmentación externa, se podría evitar permitiendo que una partición pueda fusionarse con otra, para lograr una partición nueva de más capacidad.

Esto da lugar a un nuevo esquema de manejo de memoria llamado particiones relocalizables. La idea consiste en mover celdas de memoria de un lugar a otro para juntar las áreas libres en un mismo lugar. Las celdas no se mueven, sino que sus contenidos se copian de un lugar a otro, y aunque con esto se crea un nuevo problema -el de la relocalización-, permite mayor flexibilidad que el anterior; sólo que resulta más costoso, puesto que hay que compactar (mover) los procesos a tiempo de ejecución y realizar algunos cambios en el procesador central, para evitar que este desplazamiento cause problemas con respecto a las direcciones. El procesador central se encarga de este ajuste a tiempo de ejecución por medio de un componente electrónico adicional que se conoce como registro de relocalización.

4.4.3. SWAPPING (INTERCAMBIO)

En algunos sistemas se recurre al sencillo expediente de quitar por completo de la memoria un proceso que está desactivado, copiándolo al disco magnético y liberando así un área significativa en la memoria central. Cuando llegue el momento de volverlo a ejecutar se cargará nuevamente trayéndolo del disco magnético en el que reside.

4.4.4. PAGINACION

Debido a los costos que representa la compactación (ya que es necesario detener la ejecución del proceso para efectuarla) o el swapping (porque el traslado hacia/desde el disco magnético toma tiempo), se inventó otro esquema, más ágil y eficiente, llamado paginación. Esto consiste en dividir los procesos en fragmentos de longitud fija, llamados páginas, que se almacenan en áreas de igual tamaño en memoria, llamadas bloques. Esto es, cada página de cada proceso se guarda en un bloque en memoria. Un proceso común puede constar de quince páginas, residentes en memoria en otros tantos bloques. La ventaja radica en que no es necesario que las páginas de un proceso estén contiguas en la memoria, quedando automáticamente eliminado el problema de la fragmentación externa. Con la ayuda de una tabla de mapeo de páginas, que controla cuáles páginas de qué

procesos residen en cuáles bloques de memoria, se puede implantar un esquema muy ágil de manejo de memoria central, controlado por el sistema operativo.

De acuerdo con lo dicho, si ya no es necesario que todas las páginas de un proceso estén cargadas de forma contigua en la memoria (gracias a la tabla de mapeo), entonces tampoco hay necesidad de que todas las páginas de un proceso determinado estén residentes (contiguas o no) en memoria. Es decir, se podría comenzar a ejecutar un proceso cuando tan sólo una parte del mismo esté cargada en memoria, e ir cargando a tiempo de ejecución las páginas que se requieran. Esta importante idea recibe el nombre de memoria virtual y es la base sobre la cual descansa el enorme poderío de una computadora grande, y la razón por la que una máquina puede atender a muchos usuarios al mismo tiempo aunque disponga de una memoria limitada.

4.4.5. PAGINACIÓN POR DEMANDA.

Cuando un proceso pide una página no residente en la memoria, el sistema operativo lo detecta por medio de una interrupción, que es atendida por el manejador de interrupciones del núcleo. Este determina la causa (interrupción por página) y copia la información solicitada residente en el disco magnético en un bloque libre de la memoria.

Sus ventajas son obvias, pues permite una tremenda flexibilidad en el uso de los recursos del sistema. Su desventaja es, fundamentalmente su enorme complejidad. En efecto, los sistemas operativos de este tipo constan de decenas de miles de instrucciones y se requiere un considerable auxilio por parte de los circuitos electrónicos para que la velocidad de procesamiento no disminuya radicalmente por la gigantesca cantidad de operaciones adicionales que el sistema debe ejecutar. Como la tabla de páginas reside en la memoria central, y es necesario consultarla para cada acceso, se requiere un ciclo de lectura de memoria adicional (ciclo de fetch) por cada operación sobre una página, lo cual claramente es inaceptable en términos de la reducción de velocidad de proceso resultante. Por tanto, los sistemas de paginación por demanda emplean mecanismos adicionales de hardware para auxiliarse en esta tarea. Uno de ellos es el conocido como memoria caché o memoria auxiliar rápida, en la que se guardan los contenidos activos de la sección de la tabla de páginas en uso, reduciendo grandemente el tiempo extra requerido por cada consulta. Muchos procesadores recientes trabajan en colaboración con otro complejo subsistema electrónico para el manejo de estas tareas de paginación, que recibe el nombre de unidad de manejo de memoria (MMU: memory management unit).

4.4.6. SEGMENTACION.

En éste, los procesos se dividen en fracciones llamadas segmentos. Un segmento es una unidad lógica autocontenida (un programa completo, una subrutina o un área grande de datos) que se carga en forma independiente en la memoria. La diferencia con respecto a las páginas es que aquéllas son de longitud fija, mientras que los segmentos son variables, dependiendo de la cantidad de código que contenga el programa o

subprograma que representan. El manejo de segmentos es parecido al de páginas, aunque tiene ciertas ventajas sobre éste, que ya no se estudian por tratarse de un tema especializado. Basta con saber que una máquina con sistema operativo de segmentación es por lo menos tan poderosa y compleja como otra que maneje memoria virtual por paginación.

4.4.7. SEGMENTACION-PAGINACION

Este esquema es el más complejo de todos combina las ventajas de la paginación por demanda con las de la segmentación, pero sólo lo utilizan algunas computadoras realmente grandes.

Como se ha visto, el problema del manejo de memoria es extenso y complicado, pero es fundamental para comprender lo que es un sistema operativo.

4.5 ARCHIVOS

4.5.1. SISTEMAS DE ARCHIVOS DISTRIBUIDOS

Un componente fundamental de cualquier sistema distribuido es el sistema de archivos. La tarea del sistema de archivos en los sistemas distribuidos es almacenar los programas y los datos y tenerlos disponibles cuando sea necesario.

En el caso de un sistema distribuido es importante distinguir entre los conceptos de servicio de archivos y el servidor de archivos. Por lo general, un sistema distribuido de archivos tiene dos componentes razonablemente distintos: el verdadero servicio de archivo y el servicio de directorios.

Los servicios de archivos se pueden dividir en dos tipos, según si soportan un modelo carga/descarga o un modelo de acceso remoto. En el modelo carga/descarga, que se muestra en la figura 4.13 (a), el servicio de archivo sólo proporciona dos operaciones principales: la lectura de un archivo y la escritura en un archivo. La primera operación transfiere todo un archivo de uno de los servidores de archivos al cliente solicitante. La segunda operación transfiere todo un archivo en sentido contrario, del cliente al servidor. Así, el modelo conceptual es el traslado de archivos completos en alguna de las direcciones. Los archivos se pueden almacenar en memoria o en un disco local, como sea necesario.

El otro tipo de servicio de archivos es el modelo de acceso remoto, que se muestra en la figura 4.13 (b). En este modelo, el servicio de archivos proporciona un gran número de operaciones para abrir y cerrar archivos, leer y escribir partes de archivos, moverse a través de un archivo (LSEEK), examinar y modificar los atributos de archivo, etc. Mientras en el modelo carga/descarga el servicio de archivos sólo proporciona el almacenamiento físico y la transferencia, en este caso el sistema de archivos se ejecuta en los servidores y no en los clientes. Su ventaja es que no necesita mucho espacio por parte de los clientes, a la vez que elimina la necesidad de transferir archivos completos cuando sólo se necesitan una pequeña parte de ellos.

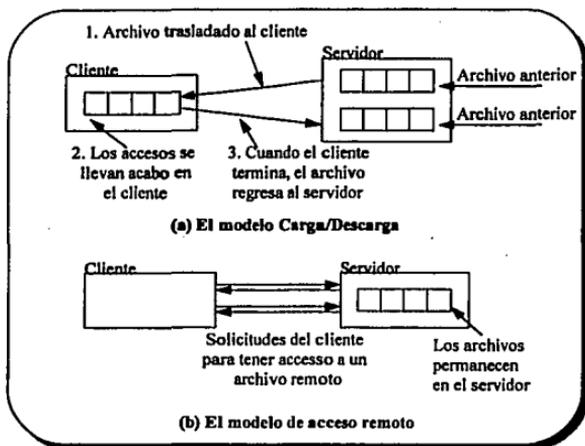


Figura 4.13 Modelos de servicio de archivos

4.5.2. LA INTERFAZ DEL SERVIDOR DE DIRECTORIOS.

La otra parte del servicio de archivos es el servicio de directorios, el cual proporciona las operaciones para crear y eliminar directorios, nombrar o cambiar el nombre de archivos y mover éstos de un directorio a otro.

Todos los sistemas distribuidos permiten que los directorios contengan subdirectorios, para que los usuarios puedan agrupar los archivos relacionados entre sí. La figura 4.14 (a) muestra un árbol con cinco directorios.

En ciertos sistemas, es posible crear enlaces o apuntadores a un directorio arbitrario. Estos se pueden colocar en cualquier directorio, lo que permite construir no sólo árboles, sino gráficas arbitrarias de directorios, que son más poderosas. La distinción entre árboles y gráficas es de particular importancia de un sistema distribuido.

La naturaleza de la dificultad se puede ver en la gráfica de directorios de la figura 4.14 (b). En ésta, el directorio D tiene un enlace con el directorio B. En una gráfica, se permite la eliminación de un enlace mientras exista al menos otro. Mediante un contador de referencias, el cual se muestra en la esquina superior derecha de cada directorio de la figura 4.14 (b), se puede determinar si el enlace por eliminar es el último.

Existen tres métodos usuales para nombrar los archivos y directorios en un sistema distribuido:

1. Nombre máquina + ruta de acceso, como /máquina/ruta o máquina_ruta.

2. Montaje de sistemas de archivos remotos en la jerarquía local de archivos.
3. Un único espacio de nombres que tenga la misma apariencia en todas las máquinas.

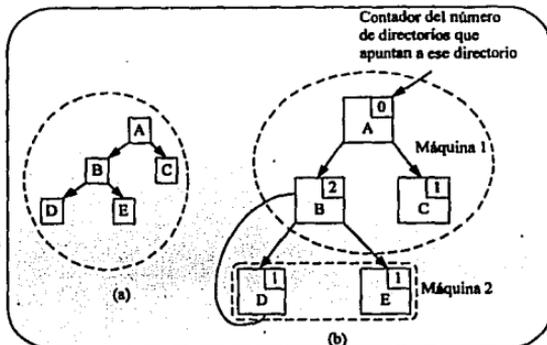


Figura 4.14 a) Un árbol de directorios contenido en una máquina. b) Una gráfica de directorios en dos máquinas

4.5.2. UN EJEMPLO: EL SISTEMA DE ARCHIVOS ANDREW.

LA ARQUITECTURA DE AFS (Andrew File System).

La configuración del AFS se muestra en la figura 4.15. Consta de unidades de asignación, con un servidor de archivos y varias docenas de estaciones de trabajo clientes por cada unidad de asignación. La idea es lograr que la mayor parte del tráfico sea local en una unidad de asignación, para reducir la carga en la columna vertebral.

Desde el punto de vista físico, no hay distinción alguna entre las máquinas cliente y servidor; todas ellas ejecutan versiones modificadas (un poco diferentes entre sí) del sistema operativo UNIX de Berkeley, con su enorme núcleo monolítico. Sin embargo, por arriba del núcleo, los clientes y servidores ejecutan software completamente distinto. Las máquinas clientes ejecutan manejadores de ventanas, editores y demás software estándar en UNIX, mientras que cada servidor ejecuta un único programa, llamado vice, el cual maneja las solicitudes de archivos de sus clientes. Cada cliente tiene además una parte de código llamada venus, que controla la interfaz entre el cliente y vice.

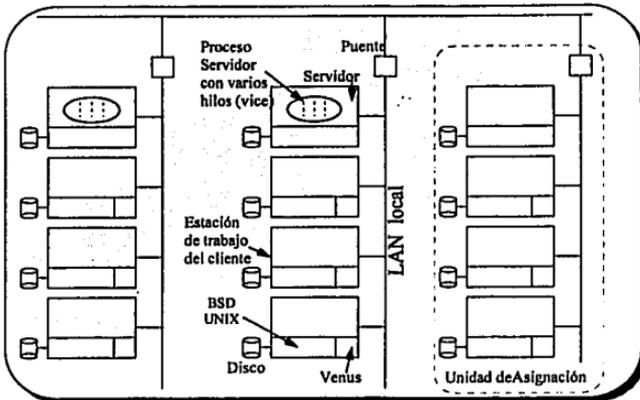


Figura 4.15 Configuración del sistema utilizado por AFS.

El espacio de nombres visible para los programas del usuario se ve como un árbol tradicional en UNIX, pero al cual se añade un directorio /cmu, como se muestra en la figura 4.16 AFS soporta el contenido de /cmu mediante los servidores vice que son idénticos en todas las estaciones de trabajo. Los demás directorios y archivos son estrictamente locales y no se comparten.

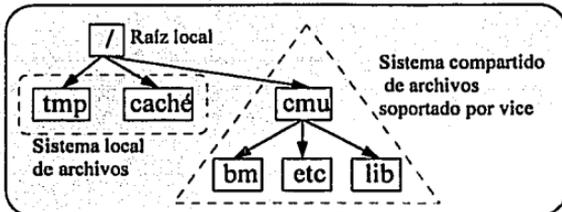


Figura 4.16 Visión del sistema de archivo desde una estación de trabajo del cliente

La idea fundamental detrás de AFS es que cada usuario realice lo más posible en su estación de trabajo y que interactúe con el resto del sistema tan poco como sea posible. Al abrir un archivo, todo éste (o si es un archivo de gran tamaño, una gran parte de él) se carga en disco de la estación de trabajo y se guarda en un caché, sin que el proceso que ejecutó el OPEN se de cuenta de ello. Por esta razón, cada estación de trabajo cliente tiene un disco duro.

Después de cargar un archivo, se le inserta un directorio local /caché, de modo que parezca un archivo normal para el sistema operativo. El descriptor de archivo que regresa la llamada al sistema OPEN designa a

dicho archivo, de modo que las llamadas READ Y WRITE funcionan de la manera usual, sin utilizar vice o venus. En otras palabras, aunque el código del sistema operativo que maneja la llamada OPEN se ha modificado en gran parte, para que maneje la interacción entre el cliente, el caché y el servidor de archivos, no se ha tocado el código de READ Y WRITE. Sólo se utiliza el archivo local en /caché, ignorando el resto del mundo.

IMPLANTACION DE AFS.

Aunque los programas de aplicación ven un espacio de nombres a la manera tradicional de UNIX, desde el punto de vista interno, vice y venus utilizan un esquema de nombres completamente distinto. Utilizan un sistema de nombre con dos niveles, en el que la búsqueda de rutas de acceso en los directorios produce entradas llamadas fids (file identifiers, identificadores de archivos). En la figura 4.17 se encuentra un fid.

0	32	0	32	0	32
Numero de Volumen	No. de nodo nodo_U		Numero Unico		
Identifica de manera, unica un volumen especifico	Indica un archivo particular en el volumen		Permite utilizar un Fid para otro archivo en el futuro		

Figura 4.17 Un Fid

Cuando un programa de aplicación llama al sistema OPEN, venus lo captura y verifica primero si el nombre de la ruta de acceso comienza con /cmu. Si esto no ocurre, el archivo es local y se procesa de manera usual. Si el nombre comienza con /cmu, el archivo es compartido. Se analiza el nombre y se busca componente a componente hasta encontrar el fid. Venus utiliza entonces el fid para verificar el caché, lo cual conduce a las siguientes tres posibilidades:

1. El archivo está presente y es válido.
2. El archivo está presente y no es válido.
3. El archivo no está presente en el caché.

En el primer caso, se utiliza el archivo en el caché. En el segundo, venus pregunta a vice si el archivo ha sido modificado desde que se cargo. El hecho de que no sea válido indica que la estación de trabajo falló y volvió a arrancar o bien que otro proceso abrió el archivo para escritura, pero en ninguno de estos casos se ha modificado el archivo y cargado en el servidor. Si el archivo no ha sido modificado, se utiliza el archivo en el caché. Si ha cambiado, carga una nueva copia. En el tercer caso, también se carga de nuevo el archivo. En los tres casos, el resultado final es que en el disco local se encuentra una copia del archivo, en /caché señalada como válida.

Las llamadas a READ y WRITE por parte de un programa de aplicación no son recibidas por venus. Se manejan de la manera usual. Las llamadas a CLOSE son registradas por venus, que verifica si el archivo ha sido modificado, y en tal caso, lo carga en el servidor que maneja su volumen.

Además de ocultar archivos, venus también tiene un caché que asocian los nombres de las rutas de acceso con los fids. Así, cuando hay que analizar una ruta de acceso para obtener el fid, se puede hacer una rápida verificación para ver si el nombre de la ruta de acceso está en el caché. En ese caso, se evita todo el procedimiento de búsqueda y se utiliza el fid del caché. Existe un problema si el archivo se eliminó después de guardar el fid en el caché y ser reemplazado por otro archivo. Sin embargo, el nuevo archivo tendrá un campo número único distinto, por lo que se detectará que el fid no es válido. Venus maneja esto eliminando la entrada (ruta, fid) del caché y volviendo a analizar el nombre a partir de cero. Si se llena el caché del disco, venus elimina archivos de él mediante el algoritmo LRU.

Vice se ejecuta como un único programa con varios hilos en cada una de las máquinas servidor. Cada hilo maneja una solicitud. El protocolo entre venus y vice utiliza RPC y se construye de manera directa en IP. Se dispone de comandos para el movimiento de archivos en ambas direcciones, cerradura y eliminación de la cerradura de archivos, manejo de directorios y otras cosas. Vice mantiene sus tablas en la memoria virtual, por lo que no tiene que preocuparse por su tamaño. El sistema operativo subyacente los página hacia dentro o hacia fuera de la memoria conforme sea necesario.

4.6 SISTEMAS OPERATIVOS DE RED

El hablar de todos los sistemas operativos significaría tratar de hacer historia de todos ellos, debido a que se han desarrollado en forma rápida. Los sistemas operativos se encuentran tanto en la parte cliente como en el servidor. En la parte cliente se encuentra un sistema operativo local como DOS, UNIX o Windows NT y en la parte servidor se utilizan DOS o UNIX sólo para levantar la máquina y posteriormente son cargados los sistemas de red como son Novell, LAN Manager, OS/2 o Windows Server sólo por mencionar los más comerciales. En la presente sección se hablará de Windows NT, debido a que este es el S.O. que de alguna manera ha sido fabricados retomando todas las características primordiales de los antes mencionados.

4.6.1 WINDOWS NT

Una de las características más importantes es la forma de construcción de su Núcleo o Kernel (o el NT Ejecutivo), esto permitirá tener varios sistemas operativos corriendo sobre NT.

CARACTERISTICAS

- Proceso distribuido: esta pensado para ambientes de red. Es capaz de distribuir tareas a otras computadoras en la red dando al usuario mayor poder, que si sólo contara con una máquina.
- Compatibilidad: Permite total compatibilidad con los actuales paquetes de software de Microsoft, esto incluye interfaces con usuario final y API's (Application Program Interface).

- **Escalabilidad y multiproceso:** Las aplicaciones son capaces de aprovecharse al máximo el rango de plataformas de computo existentes. Los usuarios deben de ser capaces de correr sus aplicaciones en máquinas con uno o más procesadores.
- **Robustez:** El sistema se protege a si mismo tanto del mal funcionamiento interno como externo. Cuidando que ninguna aplicación afecte negativamente la operación del sistema.
- **Extensibilidad:** El código crece y cambia en forma simple, según demandas del mercado.
- **Portabilidad:** El código puede viajar fácilmente de un procesador a otro. (actualmente ya abarca la familia Intel de 80X86 y algunos procesadores Risc de MIPS y Digital).
- **Seguridad**
- **Rendimiento (Performance):** es tan rápido como sea posible para cada plataforma.

ARQUITECTURA DE WINDOWS NT

Windows NT esta constituido de un Ejecutivo privilegiado y un conjunto de servidores (entiéndase procesos) no privilegiados, llamado subsistemas protegidos.

- **Modo privilegiado:** este se refiere al modo de operación del procesador, en el cual un programa puede hacer uso de todas las instrucciones de la máquina y tiene accesible el sistema de memoria. Este es llamado "Ejecutivo NT".
- **Modo no privilegiado:** en este ciertas instrucciones son deshabilitadas y el sistema de memoria no se puede acceder. Este es llamado "Modo Usuario".

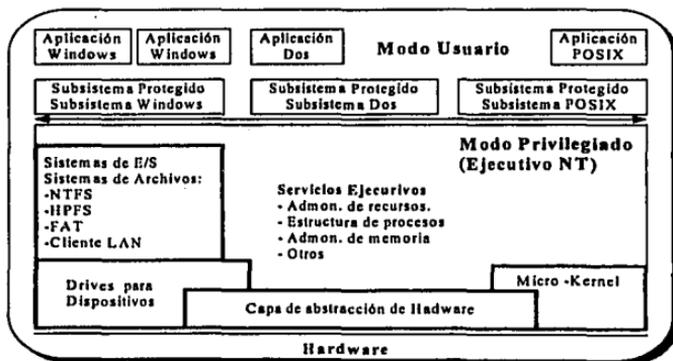


Figura 4.18 Arquitectura de Windows NT

El diseño de Windows NT permite la existencia de subsistemas protegidos (en el modo usuario) como si fueran aplicaciones que puedan modificarse o incrementarse sin afectar la integridad de otros, permitiendo con esto un sistema extensible.

Tomando en cuenta la explicación anterior de los modelos y diseños de sistemas operativos, Windows NT toma lo mejor del modelo por capas y del modelo Cliente/Servidor. En el modo Kernel de Windows NT que se conoce como NT Ejecutivo, se realizan operaciones tales como administración de memoria virtual, administración de recursos (Objetos), sistemas de archivos y de E/S, comunicaciones interprocesos y una parte de la seguridad del sistema. La mayor parte de estos componentes interactúan con otros a través de una estructura modular de capas, realizándose llamadas por medio de un conjunto de funciones internas. El modelo por capas aparece también en las partes bajas del ejecutivo en el Kernel y el HAL (Hardware Abstraction Layer), que se explicará más adelante.

Windows NT utiliza el modelo Cliente/Servidor primordialmente para proveer los servicios de API's y los servicios comunes de un sistema operativo. En la fig. 4.19, el servidor Win32 provee al usuario una interfaz fundamental hacia la operación del sistema (los términos de subsistema y servidor pueden ser utilizados indistintamente). Los otros servidores pueden ser cargados si se desea sobre el NT Ejecutivo. La comunicación entre estos subsistemas es a través de paso de mensajes usando los servicios con las que cuenta el Ejecutivo.

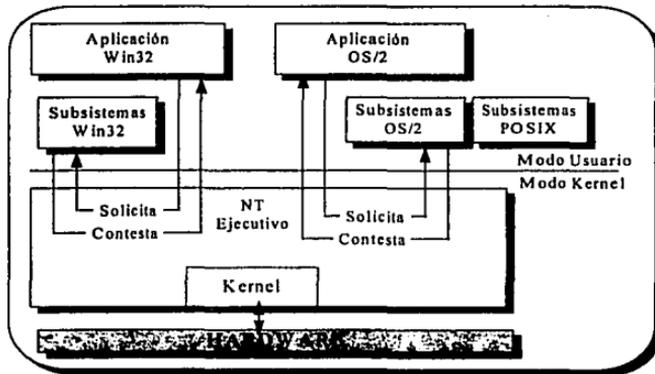


Figura 4.19

La utilización del modelo Cliente/Servidor tiene varias ventajas:

- Simplifica la operación de la base del sistema operativo. Coloca cada conjunto de API's (Win32 para aplicaciones Windows de 32 bits, MS-DOS, Windows de 16 bits, POSIX Portable Operating

System Interface for UNIX), en servidores separados libera de conflictos y duplicaciones para la base del NOS y hace más fácil agregar nuevos API's.

- Permite implementarlo con efectividad, ya que cada proceso-servidor al correr en un modo usuario tendrá su propia proporción de memoria protegida de otros procesos y como ya se comentó estos modos no pueden acceder directamente el Hardware o modificar la memoria.

Por sí mismo tienen un esquema de proceso distribuido, esto facilita que se incorpore a un esquema de red ya que el manejo de mensajes con máquinas remotas es natural y el cliente no requiere el saber que un X servicio es remoto o local en un dado caso. Para que exista una compatibilidad de software depende de varias cosas: la principal es la arquitectura del nuevo procesador, si este usa el mismo conjunto de instrucciones y el mismo esquema de direccionamiento de la memoria que el anterior, entonces existe compatibilidad binaria.

La compatibilidad binaria entre procesadores es diferente dependiendo de cada fabricante, en caso de incompatibilidad, se puede realizar con un programa emulador que convierta las instrucciones que van hacia la máquina que tenemos realmente. Si este emula todas las aplicaciones que se deseen emigrar se deberán recompilar y religar para que funcione.

CARACTERISTICAS ESPECIALES DE WINDOWS NT

Windows NT brinda soporte para Win32API, que es la interfaz de programación principal. Para equipos con procesadores Intel, un subsistema protegido ofrece compatibilidad binaria para aplicaciones MS, incluyendo MS-DOS, Windows 3.X (Windows de 16 Bits), OS/2 y Lan Manager. En un procesador RISC, la compatibilidad binaria para aplicaciones de MS-DOS y Windows de 16 Bits se realiza a través de un emulador, la compatibilidad de código fuente es ofrecida para OS/2 e interfaces de programación Lan Manager.

Windows no sólo incluye compatibilidad a nivel de API's, si no también a nivel de archivos, para lo cual Windows NT incluye soporte para los sistemas de archivos de MS-DOS (FAT), el de OS/2 (HPFS), el sistema de archivos de CD-ROMS (CDFS), y el nuevo, recuperable sistema de archivos Windows NT.

Las multitareas comparten en un procesador varios procesos, pero cuando se tienen varios procesadores se pueden compartir los procesos entre ellos. Para tal efecto existen dos modelos, el simétrico y el asimétrico. El modelo simétrico de multiprocesamiento (ASMP), típicamente selecciona uno de los procesadores para cargar el sistema operativo, mientras los otros procesadores corren otras.

Los Sistemas Operativos de multiproceso simétrico (SMP) como Windows NT, permiten que el Sistema Operativo corra en cualquier procesador disponible o en varios a la vez compartiendo memoria entre ellos. Esto permite un aprovechamiento al máximo de los recursos disponibles en el equipo.

La forma en que Windows NT implementa el multiprocesamiento simétrico, es distribuyendo partes del código del sistema operativo en varios procesadores, o varias partes en uno, a excepción del Kernel el cual debe

estar en un sólo procesador, ya que maneja las interrupciones y el sheduler; el resto del sistema puede ser distribuido o priorizado.

Para el control de errores Windows NT tiene una estructura manejadora de excepciones que captura las condiciones del error y responde a ellas de forma uniforme sin afectar a los procesos no involucrados. Windows NT controla excepciones provocadas tanto por software como por hardware como podrían ser llamadas a segmentos de memoria protegidos o divisiones entre cero; para estos existe un filtro que decide que tan grave es la falta o si el programa se culmina, continúa o se realiza otra operación.

Windows NT esta escrito en lenguaje C, el lenguaje ensamblador sólo es utilizado en las partes del sistema que están en la comunicación directa con el hardware y para aquellos componentes que requieren de una velocidad óptima. Por estas características se dice que es un diseño para ser potable.

Las secciones no portables fueron cuidadosamente aisladas con los componentes que los utiliza.

El Kernel de Windows se desarrollo para cada tipo de procesador, el cual se utiliza para todas las plataformas basadas en ese procesador. Adicionalmente a la dependencia del procesador, la dependencia de plataforma es otro concepto que Windows NT encapsula en un código de dependencias de plataforma dentro de un DLL conocido HAL o Hardware Abstraction Layer.

Windows NT utiliza un mecanismo de paso de mensajes de alta velocidad en las llamadas al sistema, llamado Local Procedure Call (LPC), que se encuentra incluido como parte del sistema.

ESTRUCTURA GENERAL DE WINDOWS NT

Este se divide en 2 partes: la parte del usuaria del sistema, la cual consiste de los Procesos_Servidores Windows NT, y la porción Kernel, llamada Ejecutivo. Los servidores Windows NT son también son llamados subsistemas protegidos, por que cada uno cuenta con su propia memoria protegida de otros procesos por medio del manejador de memoria virtual del Ejecutivo, como ya se menciono anteriormente. En vista de que los servidores no comparten memoria la comunicación entre ellos se realiza por medio del paso de mensajes, representados por flechas en la figura 4.20. El ejecutivo esta diseñado para soportar cualquier número de procesos servidores que a su vez presenten distintas interfaces de programación (API's) para distintos ambientes de aplicación: OS/2, POSIX, MS-DOS, etc.

Existen 2 tipos de subsistemas protegidos: los subsistemas integrados y los subsistemas ambientales. Un subsistema ambiental es un subsistema en modo usuario que provee API's (ej. POSIX o Windows 16 Bits). El más importante subsistema ambiental es el de WIN32, el cual provee sus API's para aplicaciones Windows de 32 Bits, estas interfaces incluyen funciones de interfaz gráfica para el usuario y todos los controladores de E/S necesarios.

Los subsistemas integrados realizan importantes tareas del sistema operativo, como es el subsistema de seguridad, el cual tiene privilegios especiales que no se encuentran en otro servidor en modo usuario.

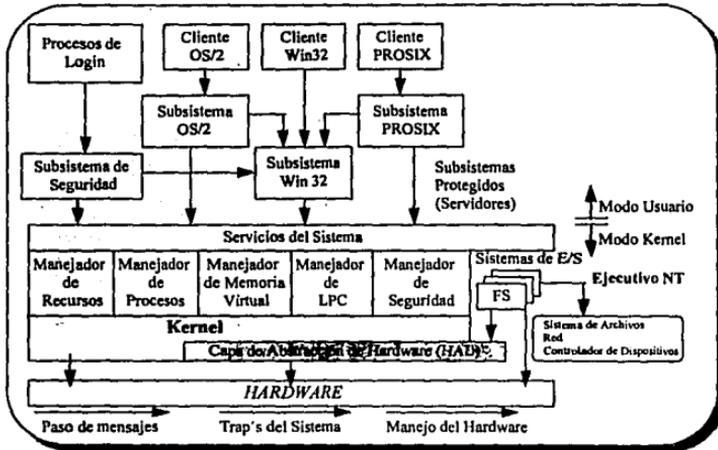


Figura 4.20 Componentes del Ejecutivo de Windows NT

COMPONENTES DEL EJECUTIVO

- **Manejador de Recursos:** Se encarga de crear, borrar y administrar los recursos del Ejecutivo.
- **Manejador de Procesos:** Crea y termina procesos, suspende o concluye la ejecución de tareas.
- **Servicio de LPC:** Pasa mensajes entre un proceso cliente y un proceso servidor en la misma computadora. LPC es una versión optimizada de lo que se conoce como RPC y que permite la comunicación entre servidores y clientes.
- **Manejador de la memoria virtual (VMM):** Implementa memoria virtual, suministrando áreas de memoria privada para cada proceso, protegiendo de invasiones el espacio asignado entre ellos. Este proceso incluye el intercambio de datos a discos (Swap) cuando es necesario.
- **Monitor de Seguridad:** Se enfoca a mantener la seguridad interna de la máquina, resguardando los recursos del Sistema Operativo y realizando auditorías en línea de cada recurso.
- **Kernel:** El cual responde a interrupciones y excepciones, realiza el Sheduler, la sincronización de múltiples procesadores, y provee de los recursos de bajo nivel necesarios para la implementación de recursos de más alto nivel.

- Sistemas de E/S: Es un grupo de componentes que se encarga de las variadas operaciones de flujo de información, entre ellos encontramos el manejador de E/S, Sistemas de Archivos, manejadores intermedios, manejador de cache, etc.
- Capa de Abstracción de Hardware (HAL): Es la capa de código entre el ejecutivo y el Hardware escrita en ensamblador y sus funciones son los detalles dependientes del Hardware como interfaces de E/S, control de interrupciones, comunicación entre procesadores, etc.

FUNCIONAMIENTO DE WINDOWS NT

Existen 2 tipos de Windows NT:

Windows NT 3.1: Este contiene servicio de red punto a punto como Windows for Group y pretende ser Windows NT para estación sencilla de trabajo sin decir que es monousuario.

Windows NT Advance Server: Este provee extensiones para entrar en un ambiente de redes controlando por dominios (redes grandes orientadas a manejar muchos servidores en un sólo ambiente), la plataforma necesaria para interactuar con otros ambientes y mejorar la tolerancia a fallas,

Para entender como Windows NT implementa sus servicios de red se hará referencia a la fig. 4.22 correspondiente al sistema OSI:

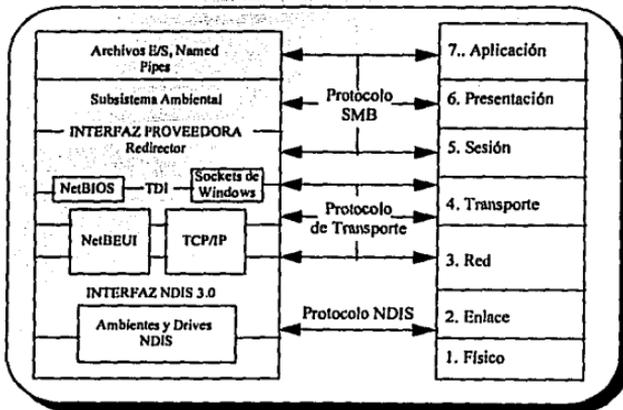


Figura 4.21 Windows NT en el Sistema OSI

Un software que se encuentra corriendo en modo usuario hace una llamada de E/S remota a través de una llamada a los servicios de E/S nativos de Windows NT, el administrador de E/S de Windows NT crea un

paquete de solicitud o IRP (I/O Request Packet) pasando la llamada a un manejador del sistema de archivos registrando, que en el caso de Windows NT recibe el nombre de redirector. El redirector pasa el IRP a los manejadores inferiores (de la capa de transporte) los cuales pondrán el paquete en la red finalmente. Cuando el paquete llega a otra máquina con Windows NT es recibido por los manejadores de red localizados en el modo núcleo, de ahí ascenderá al sistema de archivos del servidor. Este sistema de archivos servidor o simplemente server pasará el paquete al sistema de archivos local y finalmente al dispositivo físico, tal como se muestra en la fig. 4.22.

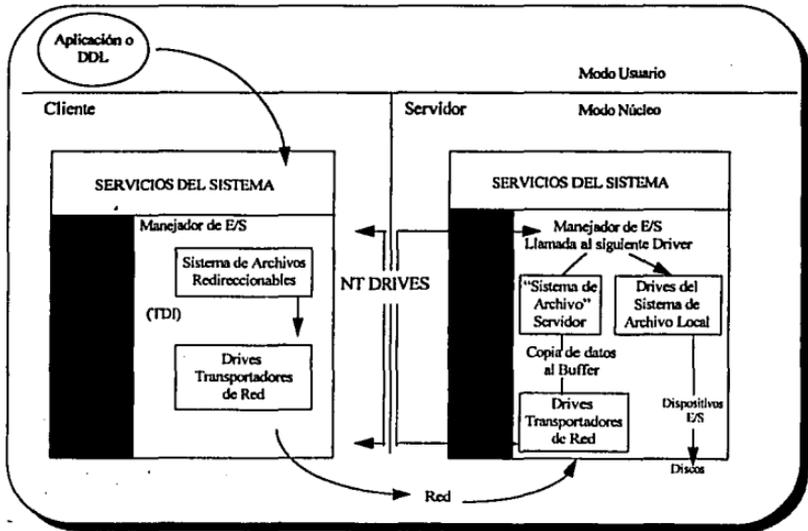


Figura 4.22 Servicios de E/S del Sistema NT

La principal ventaja del manejador del servidor de Windows NT es que esta dentro del ejecutivo de Windows NT y que puede hacer llamadas al administrador de caché directamente, optimizando la transferencia de datos, lo que repercute en un mejor tiempo de acceso.

Tanto el redirector como el Server utilizan una interfaz llamada TDI (Transport Driver Interface) para enviar y recibir los SMB's (Server Message Block) involucrados en una transacción. El TDI es un conjunto de rutinas colocadas como otro manejador en el sistema, que a su vez puede platicar con los manejadores de transporte cargados dentro de la máquina con Windows NT.

Cualquier sistema corriendo en ambiente Lan Manager podrá trabajar con Windows NT debido a que los directores son compatibles.

El diseño de Windows NT permite a terceros desarrollar en casi todos sus niveles. Windows NT soporta el acceso a distintos sistemas de archivos a través de una librería llamada Wnet API. Permite que diferentes protocolos sean cargados al mismo tiempo (Por ejemplo, TCP/IP, NetBEUI, etc.) y que estos llamen a la misma tarjeta de red. El hecho de que el acceso a tarjetas de red sea a través de drives NDIS (Network Driver Interface Specification) garantiza que las tarjetas podrán ser accedidas por varios protocolos, permitiendo a los fabricantes tener un estándar de referencia para crear sus manejadores.

Windows NT provee la plataforma necesaria para crear y correr aplicaciones distribuidas. Para la implantación de esta tecnología, Windows NT incluye el servicio de RPC (Remote Procedure Call), este servicio permite a los programadores desarrollar aplicaciones que tengan llamadas a procesos locales y a procesos remotos.

En comunicaciones remotas Windows NT Advanced Server provee un servicio incluido llamado RAS (Remote Access Server), por medio del cual se pueden utilizar enlaces de X.25, líneas telefónicas conmutadas o privadas para comunicarse.

Entre las cualidades del nuevo NTFS, destaca la completa interoperabilidad con los anteriores, rápida recuperación de errores después de que el sistema ha fallado, capaz de controlar hasta 17 billones de gigabytes, cuando se graba un dato al disco se revisa que sea lo mismo que se tenga en memoria, en caso de no corresponder se declara dañado el sector y se busca otro, a esto se le conoce como Hot/Fix.

CAPITULO V

ADMINISTRACION

DE

REDES

CAPITULO V: ADMINISTRACION DE REDES

INTRODUCCION

La administración de redes se define como: El proceso de controlar redes de datos complejas para maximizar su eficiencia y productividad.

Un error común de muchas corporaciones es el enfocarse exclusivamente en la administración de la red y olvidar la administración de los extremos: el cliente y el servidor. Se debe administrar todo clientes, servidores y red. La administración de redes ha llegado a ser un aspecto muy importante de la conectividad. Ya que las redes continuamente crecen en tamaño, recursos y complejidad. La tarea del administrador ha llegado a ser cada vez más difícil.

Todos los recurso de la red deben ser "gobemados" por diferentes servicios:

1. Administración de redes. Lo que involucra fallas, configuración, operación, seguridad, desempeño, contabilidad, activos y cambios en el sistema .
2. Administración de sistemas. Incluye administración de la red agregado a la coordinación de instalación de sistemas, coordinación de los datos generados, optimización del uso de todos los recursos, planeación, inventario, control y manejo de las finanzas. La administración de sistemas es el término usado por OSI preferentemente, reemplazando al término administración de red.

El objetivo de los usuarios de redes de computadoras no es operarlas, sino obtener de ellas la productividad y eficiencia que hagan a las empresas competitivas. De esta forma quien realiza la operación y administración de la red garantiza el rendimiento y funcionalidad de la red al máximo.

5.1 ADMINISTRACION DE SISTEMAS.

5.1.1. OBJETIVOS FUNCIONALES

El manejo efectivo de sistemas complejos demanda progresos en los siguientes aspectos:

- Procesos de administración abiertos: Los procesos de la administración de sistemas deben ser orientados a tareas, en lugar de orientarlos a productos; por consiguiente las interfaces deben ser independientes de productos y amigables.
- Datos consistentes: La definición de los elementos del sistema debe ser realizada una vez y almacenarse en un común donde se pueda tener una amplia visión de la empresa. Los datos operacionales para un subsistema o subred tienen que ser una subunidad, pero además deben ponerlos fácilmente a disposición del personal de sistemas de la empresa.

- **Administración de todas las aplicaciones en la empresa:** La administración de aplicaciones tiene que dirigirse a todos los subsistemas y subredes vitales. Se incluye desde la instalación central, departamentos de rango medio y estaciones de trabajo en el escritorio
- **Productos para la administración:** Los productos diseñados para la administración deben facilitar el manejo del estado de funcionamiento, ayuda por medio de indicadores, detección/predicción de fallas, reconfiguración sin perturbación, seguridad consistente, facilidades de seguimiento y sincronización de tiempo de eventos. (Se tratara más adelante en este capítulo el tema)
- **Automatización:** Los programas para la automatización ya sea desarrollados por el usuario o de distribuidores deben tener facilidad de uso, divulgación, interfaces programables, con asistencia de herramientas basadas en sistemas expertos

5.1.2 METODO PARA LA ADMINISTRACION DE SISTEMAS

El núcleo de la administración de sistemas es una arquitectura abierta a multiprotocolos.

FUNDAMENTOS EN LA ADMINISTRACION DE SISTEMAS

La administración de redes (especialmente redes grandes y complejas) involucra seis pasos simples:

1. **Colectar** información acerca del estado de la red y componentes del sistema (hardware y software):
 - **Medios de información definidos:** Esto incluye eventos, atributos de objetos y acciones de operación. Muchas de estas definiciones deben ser estandarizadas.
 - **Instrumentación:** Los componentes en toda la red deben ser minuciosamente instrumentados para ofrecer los eventos necesarios, alarmas, estadísticas y respuesta a comandos de operación
2. **Transformar** esta información en formatos estructurados:
 - **Protocolos de sistemas de administración:** Las reglas y procedimientos para coordinar el manejo del sistema y procesos de agentes.
 - **Sintaxis de sistemas de administración:** Las reglas para codificar y decodificar la información y protocolos
3. **Transportar** esta información a uno o más puntos para:
- 4-6. **Almacenamiento, análisis y acciones** por medio de rutinas automatizadas o con operador
 - Ofrecer una visión dinámica de la red (lógica y física).
 - Correlacionar y coordinar eventos
 - Acción automatizada donde sea posible

El análisis y las acciones son los ingredientes claves que nos llevan a los procesos de administración en vez de una mera colección de datos.

ARQUITECTURA DE ADMINISTRACION DE SISTEMAS ABIERTOS

Una administración de sistemas abiertos significa, primero que todo, ofrecer soporte administrativo para los principales protocolos de red, como APPN, OSI, TCP/IP, DECnet y NetBIOS. También significa integración de submanejadores de segmentos del sistema como LAN's y otras partes físicas de la red.

De acuerdo con lo anterior la administración de un sistema abierto debe ser creado remarcadamente a las necesidades del cliente. Debe permitir por ejemplo:

- Alertas y mensajes definidos por el usuario.
- Respuestas y acciones a mensajes definidos por el usuario.
- Diseño del usuario de filtros de mensajes antes que sean enviados o recibidos por un administrador de sistema de alto nivel.
- Aplicaciones escritas por el usuario, para recibir mensajes y enviar alertas y/o mensajes a comandos de procesos escritos por el usuario en los administradores centrales
- Alteración del usuario a la presentación de ayudas y alertas.

5.2 ESTRUCTURAS DE LA ADMINISTRACION DE SISTEMAS

Varios de los principios básicos de una estructura para la administración de sistema abiertos, multiprotocolo, son:

1. Un objeto administrado es independiente de los protocolos de comunicación.
2. Un agente común es requerido en cada nodo para servir a los objetos en ese nodo.
3. Cada subsistema de comunicación debe soportar la transmisión de información administrativa entre los servicios de administración distribuidos.

Hay tres términos para sistemas de administración de acuerdo con OSI

En el nivel más bajo tenemos *objetos*. A nivel de nodo tenemos *agentes*. Y al nivel de la administración de la región tenemos *sistema de administración*.

5.2.1. ESTRUCTURA MULTIDOMINIO

El modelo de sistema de administración que se presenta permite sistemas de administración jerárquicos. Por ejemplo, se puede tener entidades para administración, en cuatro grupos, como se ilustra en la figura 5.1. El *sistema administrador de la empresa*, junto con sus operadores (humanos y programas), representa el nivel en el cual las decisiones del sistema multidominio son tomadas. Este es el punto de control centralizado para la administración general. Un *sistema administrador de dominio* es similar, pero tiene responsabilidad sólo para su porción de la red. La responsabilidad para funciones tales como administración de fallas, seguridad, etc. pueden ser particionadas en estos dominios. La segunda capa es algunas veces necesitada para sistemas de administración anidados o *puntos de colección* con el propósito de coleccionar información desde las estructuras más bajas. En el nivel más bajo de administración, un agente maneja los elementos básicos de la red como los

componentes en el extremo de un nodo o combinación de módems y multiplexores. *El punto de servicio* es un agente especializado.

Un sistema administrador es un conjunto de programas localizados en forma central o regional a los cuales son dirigidas las acciones de requerimiento de mensajes. Los sistemas administradores pueden ser anidados. Un administrador puede, por ejemplo, ser un sistema administrador empresarial o el administrador de un dominio dentro de la empresa (ejemplos de sistemas administradores son el S/390 NetView y el Alert manager OS/400).

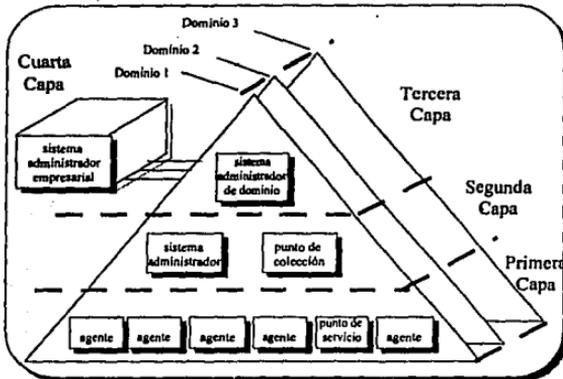


Figura 5.1 Posible estructura para la administración, mostrando cuatro grupos de sistemas administrativos

Un agente es el conjunto de programas que coleccionan información en un nodo o elemento de red (objeto) para el sistema administrador, realiza el grado de administración apropiado a este nivel y envía información hacia un sistema administrador remoto. Filosóficamente, la automatización de la administración debe ser aquí, poniendo las facilidades de la automatización tan cerca del problema como sea posible.

Un punto de colección es otro nombre para un sistema administrador anidado. Actúa como un agente con respecto al sistema administrador de más alto nivel, pero un agente que tiene responsabilidad para con otros agentes. Como un intermediario, sirve como sistema administrador hacia abajo y un agente hacia arriba. Se hace notar que la diferencia entre un sistema administrador de dominio y un punto de colección radica en la extensión de las aplicaciones administradas presentes. Por lo tanto un NetView, Alert Manager OS/400, LAN Network Manager o un OSI/Communications Subsystem Manager pueden ser vistos ya sea como un sistema administrador de dominio o punto de colección dependiendo de la extensión de las aplicaciones administradas.

Un punto de servicio es otro nombre para un agente especializado. Es un conjunto de programas que ejecuta la función de agente para otro nodo. Por ejemplo una PC puede tener esta función para un PBX.

Por ejemplo, el NetView/PC ofrece una API que puede ser usada por el usuario para escribir aplicaciones que comuniquen productos de diferentes fabricantes que usan interfaces propietarias. Con esta capacidad estas aplicaciones pueden transformar información propietaria de administración

Puntos de administración TCP/IP. El administrador de dominio en TCP/IP es llamado Protocolo simple para la administración de red (SNMP), (tratado ampliamente en el punto 5.5 de este capítulo).

Filtro de alertas. Un sistema de administración efectivo debe permitir un balance en el control centralizado y distribuido¹. Esto requiere un diseño que especifique cual situación llamara la atención del sistema administrador empresarial o el de dominio, y cual no.

El filtro de alerta usa ciertos criterios como el nombre del dispositivo, su tipo, tipo de alerta y hasta la condición específica de la falla. La presentación de la alerta correspondiente en el sistema administrador empresarial incluirá el nombre del dominio para indicar el sistema distribuido que la originó.

5.2.2. ESTRUCTURA DEL NODO DE SISTEMA DE ADMINISTRACION

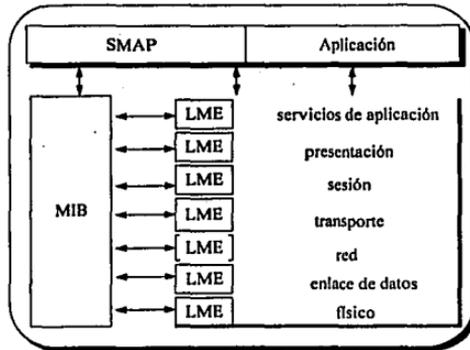
La estructura del nodo de sistema de administración OSI es mostrado en la figura 5.2 Una Entidad de Administración en Capas (Layer Management Entity) puede (opcionalmente) ser asociada a cada capa del modelo OSI invocado. Son modelos puramente conceptuales para funciones internas del sistema. Entre las funciones de la LME están:

- Colección de información administrativa.
- Reporte asíncrono de eventos.
- Cambio de parámetros operacionales.
- Cambio en el estado de operación de la capa.

Las LME's entrelazan capas individuales. Obtienen información acerca de estas capas, y efectúan cambios a esas capas. Cada LME, entonces, se comunica con un nodo Base de Información para Administración (MIB). MIB es una colección de objetos y atributos en un sistema. Este puede incluir, por ejemplo, estado de información, contadores, acercamiento a marcas de niveles y parámetros operacionales.

En cada nodo, Hay un SMAP (System Management Application-Process), el cual es la interface de MIB al mundo exterior. SMAP funciona como una aplicación y se comunica con otras SMAP's en otros nodos. SMAP, además, soporta una interface humana o programa. LME detecta errores dentro de su capa, toma acciones correctivas y notifica al SMAP cuando el número de errores ha excedido un umbral.

¹ Taber, Dave. "Alert focal-point" Interface, Abril 1989.



LME = Entidad de Administración en Capas (Layer Management Entity)
 MIB = Base de Información para Administración (Management Information Base)
 SMAP = Proceso de Aplicación de Sistema Administrativo

Figura 5.2 Funciones de un sistema administrativo en un nodo OSI

5.3 ADMINISTRACION DE APLICACIONES

El corazón de la administración de sistemas es la administración de aplicaciones. Esta abarca un muy amplio y creciente espectro de disciplinas, incluyendo fallas, configuración, cambio, operaciones, rendimiento, seguridad, costos y manejo administrativo (inventario y activos). Las categorías son algo arbitrarias, pero sirven para ofrecer una estructura. OSI se refiere a estas disciplinas como *áreas funcionales*.

5.3.1 ADMINISTRACION DE FALLAS

Este servicio es el más crítico por el costo que implica tener al sistema fuera de operación. El objetivo es determinar lo más rápido posible el punto de la red donde se presenta una falla para que esta se corrija lo antes posible, ya sea a través de la administración remota de la red o de personal de servicio que acuda al lugar donde se presentó la falla con las refacciones correspondientes. También se logra detectar fallas antes de que éstas se presenten de tal forma que el usuario nunca se entera de que estaba por presentarse una falla y que fue corregida en forma remota.

Los métodos de manejo de fallas consiste en la colección y consideración de datos a agentes en niveles bajos. Estos agentes instrumentados tiene la responsabilidad de declarar un evento cuando ha ocurrido un problema real. El reportar un evento tiene el propósito de iniciar los mecanismos de diagnóstico y recuperación.

El manejo de fallas en una sistema distribuido puede ser todo un reto. Fallas de hardware y software, y vinculaciones lógicas y físicas están involucradas. Las estadísticas deben ser almacenadas y analizadas en forma ordenada para reconocer los componentes que tienen fallas intermitentes.

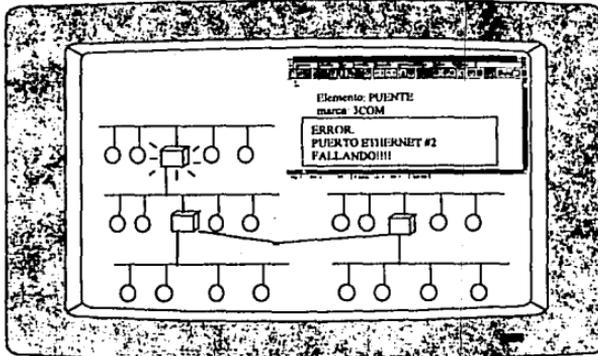


Figura 3.3 Detección y aislamiento

FASES DE UN PROBLEMA

En una estructura de red la administración de fallas maneja las condiciones de error en cinco fases:

- **Detección de la fallas.-** La detección de daños o daños retardados al disponer de un dispositivo del sistema, el aislamiento del problema por una falla de hardware, software o componentes de microcódigo.
- **Diagnostico de problemas-** Reunión y análisis de información para así determinar la causa de un problema y la acción precisa que se requiere para resolverlo.
- **Prevención y recuperación.-** Ejecución rápida de un procedimiento, automatizado hasta donde sea posible, realizado con frecuencia, para evitar el problema.
- **Solución-** La restauración de la red, restableciendo acciones preventivas, una vez que las acciones correctivas apropiadas han sido completadas.
- **Seguimiento y control.-** El seguimiento y correlación de muchos problemas hasta su resolución final. Subsecuentes análisis de desempeño y tendencias auxilian en la prevención de problemas.

PROBLEMAS LOGICOS Y FISICOS.

Involucra tres pasos: (1) la comprensión entre vinculaciones físicas y lógicas; (2) la adecuada instrumentación y mediciones que se necesiten; y (3) La información acerca de las actividades físicas y lógicas debe ser correlacionada y puesta representación gráfica individual, para así permitir la fácil navegación entre los elementos relacionados.

Los dos tipos de problemas físicos más comunes son mal funcionamiento de dispositivos o enlaces y errores en el uso de hardware tal como configuración incorrecta. Los monitores de hardware ofrecen notificación regular de ciertos eventos, alertas de mal funcionamiento y estadísticas de operaciones. El sistema

administrador analiza las alertas de mal funcionamiento de dispositivos y ofrece un conjunto de causas probables y acciones recomendadas.

La administración de fallas lógicas se refiere típicamente a problemas de software. Los dos tipos más comunes de problemas lógicos son errores de el mismo programa de software y la incorrecta definición de recursos. Si sólo una parte lógica de un nodo extremo falla, un componente en este nodo extremo generará un mensaje a su otro extremo.

5.3.2. CONFIGURACION Y ADMINISTRACION DE OPERACIONES.

Se debe controlar toda la información detallada de los elemento que componen la red, software y hardware, números de serie, versiones, etc. Asimismo se recomienda tener un mapa de configuración actualizada conforme se van añadiendo componente o se realizan cambios de este.

Esta dentro del ámbito de la operación el establecer y modificar la configuración de la red, monitorear la continua operación de sus elementos, ofrecer servicios de respaldo/archivo y auxiliar en el control de la carga de trabajo a través de la red. Por último puede incluir la programación de ejecución de tareas y trabajos a través de la operaciones distribuidas.

Las operaciones son algunas veces manejadas desde una localidad centralizada, localidad o combinación de ambas. Un grado de centralización no sólo permite el análisis un amplio contexto sino que además concentra la pericia del personal de operaciones.

CONTROL DE CONFIGURACION.

Las operaciones de configuración incluyen el inicio, terminación y reconfiguración.

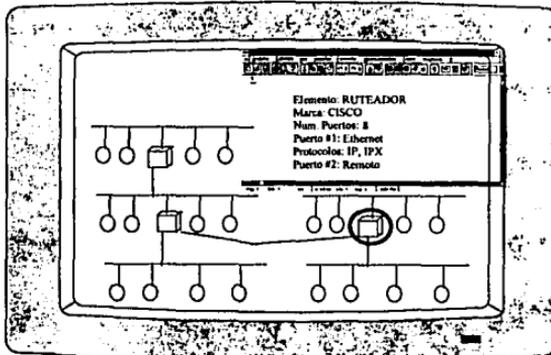


Figura 5.4 Configuración de dispositivos

Por ejemplo un operador usando una sola consola puede activar múltiples procesos, ajustar el reloj, encender dispositivos y apagarlos, inicializar sistemas operativos (sin conocer el lenguaje único de operación de cada sistema) e inicializar los principales subsistemas.

ADMINISTRACION PARA CAMBIOS

La administración para cambios se refiere a la expansión, actualización y reemplazo de software, microcódigo, características de hardware². Esto puede deberse a cambios planeados o nuevas versiones de software o módulos de microcódigo, dilemas resultado de fallas de acciones administrativas, cambios de reingeniería, etc.

Una solicitud de cambio, por ejemplo, puede requerir la coordinación de personal de operaciones y comunicaciones, gentes de seguridad y programadores de sistemas ya sea en sitios locales o remotos. La tablas de sistemas y subsistemas tiene que regenerarse. Software, hardware y/o microcódigo tienen que instalarse. También cableado y dispositivos tienen que instalarse. Se tiene que planear y evaluar las pruebas en la configuración antes y después de los cambios.

FACES EN LA ADMINISTRACION DE CAMBIOS.

Los principales componentes en la administración de cambios son:

- Planeación. El proceso de planeación del cambio comienza con un cuerpo de información total sobre los niveles actuales de hardware, software y microcódigo en las red. El plan para el cambio necesita ser fácilmente ejecutable. Esto requiere frecuentemente la capacidad distribuir automáticamente, instalación remota y seguimiento de la instalación.
- Distribución. La distribución, hecha electrónicamente³, incluye la planeación, programación y seguimiento de la distribución de datos, software y microcódigo
- Instalación. Esto involucra instalación sin operador en locaciones remotas. A causa de que los cambios no pueden ser completamente exitosos, es necesaria una capacidad rápida para regresar los recursos a su condición previa. Además los cambios son instalados sobre una base ya probada, o son programadas las actividades de instalación para una fecha y hora particular.
- Seguimiento. Al ser completada toda la instalación, la configuración de la base de datos central tiene que ser actualizada.

CONTROL DE INVENTARIO

Mantener el seguimiento de los nuevos componentes, movimientos y otros cambios en una red ampliamente distribuida puede ser tiempo consumido, determinar si los puertos están vacíos o relativamente no

² Ballard C. P., Farfara C. and Heloke B. S. "Managing changes in SNA networks", IBM system journal, vol. 8, No. 2, pp. 250-273.1989

³ Autru Pasquale and Ballard Christopher, "Netview distribution manager; an update", Interface, No. 36 pp. 45-50

usados es otra operación regular cuando están siendo encontradas nuevas demandas. Estos trabajos de labor intensa han conducido a la creación de indicadores automáticos desde los mismos elementos de la red.

5.3.3 ADMINISTRACION DEL DESEMPEÑO

Se realiza un monitoreo de la utilización de la red detectando sobrecargas que afecten el buen funcionamiento del sistema. Se analizan también las áreas donde el tráfico tiende a crecer. Con la administración del desempeño se logra eliminar una de las características más comunes en las redes instaladas la cual es que su crecimiento no fue necesariamente ordenado, lo que conduce a cargas desbalanceadas de trabajo entre servidores, grupos trabajo o sectores de la red. Con un buen análisis de rendimiento y sin necesidad de invertir en hardware o software adicional se puede incrementar el rendimiento del sistema

Así, la administración del desempeño se esfuerza por asegurar un óptimo rendimiento de la red, o al menos por ofrecer un nivel de rendimiento encomendado por los usuarios. El rendimiento desde el punto de vista del usuario es sensibilidad y disponibilidad. ejemplos de funciones de esta área son:

- Medidas de tiempos de respuesta.
- Disponibilidad de monitoreo
- Monitorear retraso de componentes

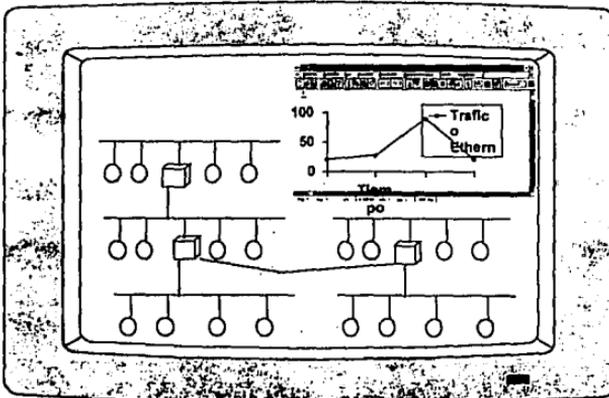


Figura 5.5 Monitoreo y análisis

5.3.4 ADMINISTRACION DE COSTOS Y USO DE LA RED

Aquí se puede establecer la tarifa del uso de la red por centro de costos o usuarios del sistema. De esta forma se pueden determinar las áreas de mayor costo de la red y este costo tiene sentido con las áreas usuarias

Los diversos servicios, distribuidos por toda la red, son vistos por los usuarios cada vez más como utilerías. Por lo tanto llega a ser necesario ofrecer un buena contabilidad de los activos de la red y servicios de uso.

Una aplicación de contabilidad ofrece, por ejemplo, datos con respecto al cargo de uso y existencia contable, para que el cliente pueda producir facturas de actividad de datos en la red. Es necesaria información suficiente sobre tráfico de voz y datos para suministrarla al departamento de facturación de acuerdo a la identificación del usuario y extensión telefónica. Como ejemplo, alguna de la información útil del tipo contable incluye:

- Inicio y fin del tiempo de conexión
- Razón del fin de la conexión
- Número de mensajes de texto enviados y recibidos
- Número de bytes de texto enviados y recibidos
- Número de producto
- Reporte del nombre de nodo
- Identificador de evento
- Nombres de los punto de acceso al servicio
- Selector de punto de acceso al servicio
- Identificación del punto de fin de conexión

Las aplicaciones de contabilidad pueden establecer limites sobre el uso de objetos administrados y puede monitorear estos umbrales. Los costos del uso de múltiples objetos puede ser combinado y pueden prepararse reportes periódicos para los usuarios.

Grupos de usuarios	Num. de paquetes enviados		
Finanzas	114,315		
Sistemas	267,890		
Administración	4	Usuarios	Tiempo de conexión
Personal	3	Hugo	3:15
Soporte	5	Ramiro	2:26
Mantenimiento	1	Alejandra	7:59
		Araceli	5:34
		Joel	3:22

Figura 5.6 Contabilidad de recursos

5.3.5 ADMINISTRACION DE SEGURIDAD

INTRODUCCION

Aquí se debe controlar el acceso solamente a cuentas autorizadas en cualquier punto de la red. Se obtienen reportes de la fecha y modo de acceso a la red. De esta forma se logra garantizar el acceso a la red sólo al permiso autorizado y se minimiza el riesgo de daño a la red.

En general una estructura de administración de seguridad en una red amplia debe contemplar mecanismos de seguridad para lo siguiente:

- Identificación del usuario y validación: Reconocer y verificar la identidad declarada por un usuario (o servidor).
- Control al acceso de recursos: Permitir el uso de recursos sólo por aquellos convenientemente autorizados
- Confidencialidad: Incluir comunicaciones seguras y el manejo de claves para encriptamiento.
- Integridad de datos: preservación de los datos ante cambios no autorizados

IDENTIDAD Y VALIDACION.

La mayoría de los sistemas computadores que existen proporcionan una forma de validación: un usuario prueba su identidad al servidor. Generalmente el usuario no necesita que el servidor pruebe su identidad. De acuerdo al crecimiento del sistema distribuido esta confianza al servidor puede ir disminuyendo, y por tanto llegaran a ser necesarias dos formas de validación.

Otra consecuencia de una distribución amplia de los sistemas es el requerimiento de que el usuario no tenga que probar su identidad a todos y cada uno de los servidores que usa. Una señal individual es deseada para validar un servicio, lo cual facilita la certificación del usuario a todos los servidores.

AUTORIZACION DE ACCESO.

La distribución de bases de datos y el fácil acceso a un amplio rango de información son objetivos primarios de muchas empresas. Esto es muy ventajoso sin embargo, tiende a exponer la seguridad.

Un número de diferentes métodos de control de acceso pueden ser usados. La mayoría de los sistemas emplean un sistema de control de acceso donde el propietario del recurso puede especificar quién puede acceder este recurso y que tipos de permisos le están permitidos.

Accesos Desagregados	
ESTACION	HORA
186	15:40
56	9:34
200	13:5

Estaciones Fisicas No Declaradas		
ESTACION	TIEMPO DE CONEXION	
192	4:58	
8	2:12	
26	5:31	

Figura 5.7 Seguridad

5.3.6 ENCRIPITAMIENTO DE DATOS

Los medio de transmisión y almacenamiento de datos también necesitan ser asegurados. Un soporte de alto nivel es necesario para encriptar sólo la porción de datos. El encriptamiento de este tipo trae un beneficio adicional, en caso de error, el proceso de desencriptar fallara y se detectara un error si aunque sea un sólo bit del mensaje esta corrompido o si un paquete que es sólo una parte del mensaje esta perdido o duplicado.

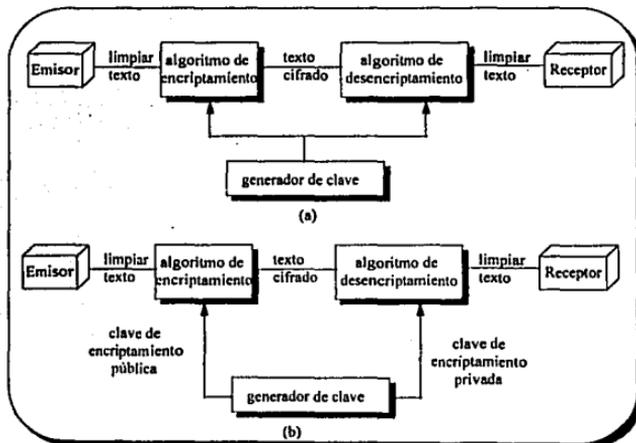


FIGURA 5.8 Dos estilos de encriptamiento: (a) encriptamiento con clave privada; (b) encriptamiento con clave pública

ENCRIPTAMIENTO PUBLICO Y PRIVADO.

Han aparecido dos tipos de encriptamiento en los estándares: la clave privada o encriptamiento simétrico y clave pública o encriptamiento asimétrico. Estos dos procesos son ilustrado en la figura 5.8

El método de encriptamiento de la clave privada transforma los datos del emisor con la ayuda de un algoritmo de encriptamiento público y una clave binaria individual que es privada, conocida sólo por el emisor y el receptor, La clave privada, por lo tanto, debe ser protegida. La técnica más popular que es usada para el encriptamiento es el algoritmo Estándar de Encriptamiento de Datos (Data Encryption Standar. DES). El algoritmo DES creado por IBM.

El método de encriptamiento de la clave pública usa dos claves, una debe mantenerse ser privada como clave propietaria y la otra puede ser pública, quizá hasta publicarse en un directorio. Ambas son generadas juntas y la clave pública no puede ser usada para derivar la clave privada. El algoritmo para el encriptamiento de la clave pública es el RSA, desarrollado por Rivest, Shamir y Adleman. Usando métodos con la clave pública, el emisor y el receptor no tiene que compartir una clave secreta en común. Sólo la clave privada debe ser protegida; si esta (y la clave pública) son generadas por el receptor, la clave privada no tiene que ser transmitida. Es posible dos diferentes modalidades:

Mensaje secreto. Se envía un mensaje secreto al receptor, el emisor encripta el mensaje bajo la clave pública. Este mensaje sólo puede ser descifrado por el receptor que tiene la clave privada. El mensaje no puede ser descifrado con la clave pública.

Para simplificar la identificación del emisor, sin ser secreto el mensaje, el emisor encripta el mensaje bajo la clave privada. Este mensaje puede ser descifrado por alguien que tiene la clave pública. El que recibe puede tener la certeza de que el emisor posee la clave privada correspondiente.

La desventaja con los sistemas de clave pública es que depende de funciones matemáticas difíciles de revertir, es difícil de llevarlo a cabo en implementaciones de bajo costo

5.4 RESPALDOS

Normalmente, es un error permitir a los usuarios tener archivos de datos dentro en sus terminales. Es muy raro que un usuario respalde su datos en forma rutinaria. Algunas organizaciones evitan este problema no poniendo ningún disco en los clientes de una red. Sin embargo, las redes con clientes sin disco pueden llegar a ser una carga pesada para el sistema con reinicios del sistema cliente. a menos que se haga un uso no frecuente del la LAN o las aplicaciones no sean cargadas frecuentemente. Las configuraciones sin disco son comúnmente poco practicas.

En algunos ambientes Unix, una estación de trabajo sin datos puede ser usada en lugar de una máquina sin disco. Una máquina sin datos es un estación de trabajo con un disco local que es usado sólo por el sistema para el cambio de pagina. Todas las aplicaciones y ejecutables del sistema residen sobre el servidor de archivos

y son cargadas en la estación de trabajo vía la red. Una configuración sin datos presenta muchos de los inconvenientes de un ambiente sin disco, especialmente si el usuario tiende sólo a acceder sólo un grupo de aplicaciones. El único golpe sobre la red es el arranque inicial de la estación de trabajo y la carga de la aplicación

5.4.1. RESPALDOS SOBRE LA RED

Algunas organizaciones permiten al usuario almacenar datos en forma local e intentar ejecutar un respaldo a través de la red. En un respaldo en la red, un servidor de respaldos copia los archivos del cliente a través de la red y los respalda. Aunque esto puede trabajar bien sobre una LAN pequeña, en grandes redes este método puede ser una carga de problemas de operación.

Por ejemplo, si un usuario apaga su terminal, no se le puede hacer ningún respaldo. O la información del respaldo puede corromperse con errores de comunicación o aun fallas debido a una caída de la red. El respaldo puede fallar debido a una caída del sistema cliente. Si alguna caída del sistema ocurre, el software de respaldo en red debe ser capaz de recuperarse y continuar.

Lo que hace una utilería de respaldo en red si enfrenta una falla es una cuestión importante. ¿Puede siempre esta detectar si una falla ha ocurrido? ¿Reintentará o abandonará? ¿Si abandona, continuará con el siguiente sistema cliente? ¿Que le pasa a los datos ya respaldados si una falla ocurre? ¿Puede ayudarnos a encontrar la causa?

Además ¿Cuantos datos hay sobre los clientes?

Un método común para acelerar los respaldos sobre la red es tratar de explotar el hecho de que muchos datos sobre el cliente no son transitorios o no cambian rápidamente. Por ejemplo, software del sistema, archivos fuentes, aplicaciones binarias y texto de ayuda puede consumir decenas de MGbytes de espacio de disco que no cambia.

Ya que los respaldos son usados ocasionalmente para restaurar datos, representa un problema el tratar de encontrar en cual dispositivo de respaldo se encuentra un archivo determinado.

Otra desventaja de los respaldos sobre la red es que pueden crear un riesgo seguro. Si un cliente esta configurado para permitir el respaldo remoto de un usuario, existe la posibilidad de que un intruso realice el respaldo.

5.4.2. RESPALDO CON SERVIDORES DE ARCHIVO

Un mejor método es el usar un servidor de archivos para almacenar todos los datos de los usuarios. Un disco local en el cliente deberá ser usado sólo para iniciar el sistema, manejo de memoria y posiblemente para almacenar algunas aplicaciones binarias.

¿Como accesa el usuario sus archivos? La solución es usar un sistema de archivos que de la apariencia de que los datos del usuario en el servidor de archivos se encuentran en forma local en el cliente. Hay varias opciones de software para realizar esto: Novel Netware, Banyan Vines, Microft LAN Manager, IBM LAN Server, Appleshare, NFS, etc.

Con esta solución el servidor de archivo es respaldado localmente. Esto elimina varios puntos de falla, reduce la carga en la red, acelera el proceso de respaldo y evita problemas de seguridad. Los respaldos también pueden fallar debido a errores en el medio de respaldo o fallas del servidor de archivo. Hay menos puntos de fallas utilizando un servidor de archivos que usando respaldos sobre la red.

La restauración de datos es además simplificada. Ya que el servidor de archivos contiene en forma primaria datos de usuario no estáticos. Es frecuente la factibilidad de realizar respaldos de todo el servidor de archivos. Esto simplifica enormemente el proceso de restauración . La figura 5.9 ilustra las diferencias entre respaldos sobre la red y el con el servidor de archivos.

Este método funciona bien para servidores de archivos, pero que hay acerca de respaldar otros servidores o bases de datos. De nuevo la opción es entre configurar estas maquinas con dispositivos de respaldo local o realizar respaldos sobre la red.

Como hay una gran posibilidad de falla, un respaldo sobre la red generalmente requiere más costos de operación. Cuando calculamos el tiempo de vida de un servidor, los costos de operación de respaldos sobre la red son usualmente más caros que los costos de agregar un dispositivo de respaldo a cada servidor.

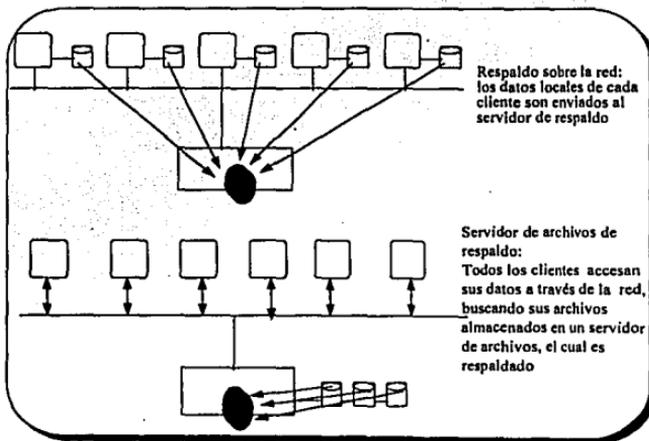


Figura 5.9 Respaldo sobre la red Vs. respaldo con servidor de respaldos

5.4.3. RESPALDOS SIN OPERADOR

En la mayoría de los casos que se realizan respaldos en ambientes distribuidos, no es usualmente factible el hacer respaldos con ayuda de un operador.

Operadores Ocasionales

Aun usando operadores momentáneos esto no es siempre práctico, ya que algunas maquinas permanecerán ociosas esperando la llegada del operador.

Algunas bases de datos y sistemas de archivos permiten hacer respaldos de respuesta rápida (hot backups), sin tomar el control del sistema. Mientras esto suena bien en teoría, la dificultad con este método es que no se sabe cual es el estado exacto del sistema cuando va a ocurrir el respaldo. Hay que recordar que una política de respaldo es tan buena como sea su capacidad de restauración. Restaurar un complejo ambiente que fue respaldado con actualizaciones en curso esta lejos de ser fácil. Son necesaria herramienta automáticas para saber cuales respaldos serán usados para restaurar. A menos que se puedan repetir todas la actualizaciones que ocurrieron después del punto en que el respaldo fue tomado. Un respaldo de respuesta rápida estará por un tiempo largo sin uso.

5.4.4. ADMINISTRACION DE RESPALDOS

Los respaldos automáticos presentan sus propios problemas. ¿Que hará el software de respaldo en caso de presentarse una falla?

Una clave para un respaldo exitoso es asegurarse de examinar los procedimientos de respaldo antes de usarlos. Si se sabe como reaccionara el software de respaldo en una gama amplia de condiciones de error, se puede asegurar que los procedimientos seguidos cubran estas consecuencias.

Un ingrediente esencial en el funcionamiento sin problemas de respaldos automáticos es llevar estadísticas correctas del software de respaldos usado. Conociendo que cantidad de datos se respaldo, cuanto tiempo tomo, que errores se encontraron, que acciones se tomaron para remediarlos, que porcentaje del medio de respaldo fue usado, etc., será información invaluable para planear y administrar en forma proactiva.

Finalmente, es importante un buen control sobre los dispositivos de respaldo. Conocer el volumen que se usa, donde se encuentra, cuando expira, cuando se uso, cuando deberá reusarse, etc.

5.5 SOFTWARE PARA LA ADMINISTRACION DE REDES

5.5.1. PRODUCTOS PARA LA ADMINISTRACION

Una estrategia de administración proactiva de red incluye monitoreo de la red utilizando niveles sobre todos los segmentos de esta. Conociendo cuales grupos de trabajo, subredes y backbones están en peligro de una sobrecarga puede ayudarnos a prevenir problemas innecesarios. Hay varios paquetes fuera del estante para la administración de redes tales como HP's Open View, Sun's SunNet Manager e IBM's Netview. Estos paquetes

pueden monitorear la utilización de niveles, y compararlos con índices previos y disparar alarmas cuando el umbral de las marcas se exceda.

Muchos errores de red son inducidos por el hardware. El continuo monitoreo puede resaltar errores transitorios resultado de fallas de hardware antes que la parte falle completamente. Si se observa que el índice de fallas de tarjetas de interface se incrementa, por ejemplo, se puede detectar que una tarjeta necesita reemplazarse antes que se comiencen a notar problemas con ella.

En casos donde hay errores relacionados con el software, se necesitará examinar el contenido de los paquetes de red y descifrar su contenido. Los más modernos ámbitos de monitoreo de red, tales como Network General's Sniffer, pueden diagnosticar los principales protocolos de red, como Ethernet, TCP/IP, Token Ring, SNA, Novell y otros. Aun cuando un paquete con esta característica es una actualización cara para el monitoreo de redes, vale la pena su inversión.

Una característica clave que se debe de buscar en las herramientas de administración de red es la capacidad de filtrar datos. Debido al alto volumen de datos en la red, es importante que sea capaz de extraer datos de interés. Debe ser capaz de incluir o excluir paquetes basándose en el recurso o su dirección destino, tipo de paquete, tipo de protocolo y paquetes inválidos o erróneos. En una red grande, un sólo error puede fácilmente generar aproximadamente unas 5,000 alarmas en un minuto. Sin una forma de filtrar las alarmas causadas por errores en cascada, encontrar el problema real puede ser como encontrar una aguja en un pajar. De manera creciente, muchos paquetes para la administración de redes ofrecen una extensión de sistemas expertos que no sólo pueden filtrar alarmas sino guiar a la raíz del problema. Un buen ejemplo de estos paquetes es el producto de Cabletron llamado Spectrum.

Uno de los grandes problemas de la administración de redes es el ofrecer acceso a los usuarios en forma eficiente y de fácil uso. En un mundo de redes de múltiples servidores, el usuario puede llegar a sentirse desesperadamente perdido tratando de determinar exactamente en cual subdirectorio de cual volumen de cual servidor de archivos esta la información que busca. Tomado en cuenta este problema los fabricantes de sistemas operativos han desarrollado software de administración para asignar nombres a los recursos en forma global (Utilizando los métodos discutidos en el capítulo 3). Steattalk un producto de Banyan Vines es un ejemplo de este tipo de software.

5.5.2. PROTOCOLO SIMPLE PARA LA ADMINISTRACIÓN DE REDES (SNMP)

Un problema con la mayoría de las herramientas existentes de administración es que son específicas del fabricante, es decir son propietarias. El surgimiento de un abanico de empresas con redes obliga a tener una mezcla heterogénea de fabricantes de redes. Las herramientas propietarias que sólo trabajan con determinados fabricantes software no llenaran esa necesidad.

Afortunadamente, la industria ha escuchado los ruegos y ha intentado resolver el problema con el desarrollo del Protocolo para la Administración de Red Simple (SNMP) de TCP/IP, un protocolo estandarizado

que hace posible mantener interredes complejas y reconfigurar cambios usando patrones. SNMP fue diseñado para el propósito de monitorear le rendimiento de una red, detectar fallas y su aparición y configurar o reconfigurar recursos de la red. SNMP ejecuta la contabilidad de tareas y ofrece capacidades para el ordenamiento de detección de fallas.

El SMNP esta definido en una serie de documentos llamados RFC's. Una Estación Administradora (Network Management Station, NMS) da al administrador una visión del estado de la red y un control directo sobre sus dispositivos. Los (software) agentes pueden ser localizados, por ejemplo, en ruteadores, puentes, servidores y estaciones de trabajo. Hablando estrictamente SMNP es el protocolo común para que cualquier agente, platique con la NMS.

Las especificaciones de la Base de Información para Administración (MIB) (contenidas en RFC 1155 y 1156) describen los objetos que pueden ser manejados usando SNMP, es decir, MIB es la forma común en que se ven los datos.

5.5.3. FUNCIONAMIENTO DE SNMP

SNMP define cinco verbos (comandos) para realizar sus funciones: De estos cinco verbos, tres son utilizados permanentemente en el dialogo que la NMS mantiene con cada dispositivo a través de agente que se encuentra en cada uno de ellos (ver figura 5.10)

Por ejemplo, imaginemos que se esta viendo en su NMS el mapa de toda su red, y deseamos analizar con más calma lo que esta sucediendo en un concentrador determinado. Para ver la información más a detalle señalamos con el Mouse ese concentrador, con lo que aparece en la pantalla la información específica de ese dispositivo.

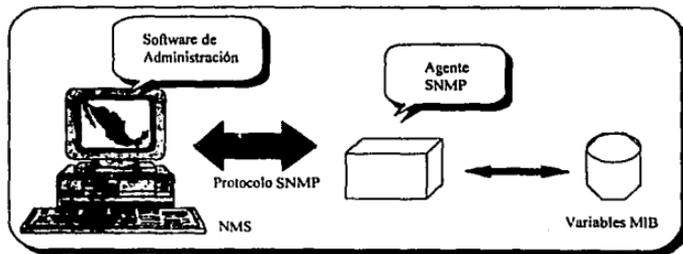


Figura 5.10 Esquema general usando el estándar SNMP

Lo anterior parece bastante automático, sin embargo para que pase todo eso tuvo que existir todo un diálogo (bajo protocolo SNMP) entre la NMS y el software agente que tiene el concentrador (suponiendo que se tiene un concentrador inteligente).

El diálogo entre estos elementos sería así:

NMS.-	Pregunta al agente ¿Cuantos puertos tienes?
Agente	Responde con el número de puertos (Ej. 12 puertos)
NMS	Pregunta ¿Cual es el status del puerto número 1?
Agente	Todo OK
NMS	Pregunta ¿Cual es el status del siguiente puerto?
Agente	Todo OK
NMS	Pregunta ¿Cual es el estado del siguiente puerto?
Agente, etc.	Todo OK

El diálogo anterior se lleva a cabo mediante los tres verbos más usuales de SNMP: *GetRequest*, *GetNextRequest* y *Get Response*.

A continuación explicaremos los cinco comandos utilizados por SNMP:

- *Pedir un dato (GetRequest)*.- Se utiliza cuando la NMS (Estación Administradora) requiere de saber el valor de cualquier variable MIB de algún dispositivo.
- *Pedir el dato siguiente (GetNextRequest)*.- Cuando se ha preguntado un dato y se desea saber el siguiente en la lista. Así por ejemplo, se pregunta por el status del puerto 2 de un concentrador Ethernet, y también se requiere saber del status del puerto 3, y del 4, etc. Estas peticiones las hacemos a través de *GetNextRequest*.
- *Poner un valor a una variable (SetRequest)*.- La NMS usa este tipo de comando cuando necesita fijar un valor a una variable de algún dispositivo.
- *Contestar a los tres mensajes anteriores (GetResponse)*.- Este mensaje lo utiliza el Agente SNMP de cada equipo de la red, para contestar a un mensaje *Get*, a un *GetNext* o a un *SetRequest*.
- *Avisar algún suceso importante del lado del dispositivo (Trap)*.- Este es el único mensaje que NO se origina en la NMS, sino en el propio dispositivo. En otras palabras, el agente SNMP que se encuentra corriendo dentro de cada elemento administrado, envía este mensaje cuando detecta alguna condición de error.

ESTRUCTURA DE VARIABLES MIB

En las ramas del árbol de la figura 5.11 se representan diferentes grupos de objetos o variables, dentro de los cuales aquellos que nos interesan están en la rama denominada "MIB".

Haciendo un recorrido desde la raíz para llegar a esa rama se comienza por la rama nombrada como "iso" (que es la número 1) y después por la rama "org" (que tiene un número 3), y así sucesivamente por "dod" (que es la 6) y a su vez por "internet" (que es 1) y finalmente por "mgmt" (número 2).

El árbol con todas sus ramas está diseñado para que pueda servir para diversas aplicaciones y tipos de datos, no sólo para administración de redes. Sin embargo, en este momento lo que interesa son las ramas nombradas como "mib".

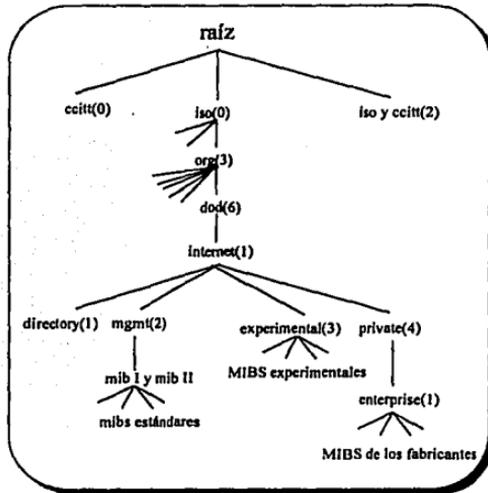


Figura 5.11 Árbol mostrando la estructura de MIBs

No obstante, resulta ser que "mib" es la primera rama (de ahí el número 1) dentro del grupo "mgmt", que a su vez es la rama dentro de internet que se encarga de todo lo relacionado a administración. Como "mgmt" es la segunda rama de internet, de ahí el número 2.

Siguiendo con la ruta seguida, internet (que agrupa todo lo relacionado con SNMP y TCP/IP) es a su vez la primera rama del grupo "dod" (o sea el Pentágono o Departamento de Defensa) que a su vez pertenece al grupo "org" (organizaciones) que finalmente es parte de "iso" (organización internacional de estándares).

Todo esto se puede sintetizar que para llegar a "mib" se tiene que pasar por los siguientes números de ramas: 1.3.6.1.2.1

En donde para evitar confusiones, se acostumbra poner puntos separando cada valor.

Y dentro del grupo "mib" a su vez hay más ramas y subramas, abarcando diferentes objetos, los cuales tienen ya una función específica (Por ejemplo un objeto podría ser el número 1.3.6.1.2.1.3.5 el cual -supongamos- podría controlar los paquetes que han llegado al dispositivo, por lo cual podría tener un nombre, digamos: Total PaquetesEntrada)

Como las variables del grupo "mib" no cubren todos los parámetros que se puedan tener para cualquier dispositivo, existen grupos adicionales, tanto en experimentación como particulares a cada fabricante.

Los MIB's experimentales (camino 1.3.6.1.3) tienen variables en grupos tales como: Ethernet, Token-Ring, FDDI, Lan Manager, etc. Los MIB's de cada fabricante (camino 1.3.6.1.3) son variables que éstos, los

fabricantes, agregan a sus equipos, para llevar el control de funciones específicas que no se encuentran en los MIB's estándares.

5.5.4. PROTOCOLO COMUN DE INFORMACION DE ADMINISTRACION (CMIP)

Se han hecho otros intentos para estandarizar los protocolos de administración, Como es el caso del Protocolo Común de Información para Administración (CMIP). CMIP (cae dentro del modelo general de administración de objetos. Desarrollado por OSI) fue proyectado para profundizar en el diseño de la distribución, redes heterogéneas y administración de aplicaciones.

CMIP tiene la capacidad de para monitorear tanto aplicaciones como dispositivos. Esto es debido a la naturaleza extendida de CMIP, el cual tiene la capacidad de agregar nuevos objetos dinámicamente para monitorearlos.

ADMINISTRACION DE OBJETOS BAJO OSI

OSI ha especificado un modelo general de administración de objetos, el cual ofrece formatos para coleccionar información comparables de una gran variedad de tipos de objetos. Como se menciono anteriormente, ejemplos de los objetos que se pueden administrar incluye objetos lógicos como entidades en las capas (aplicaciones), y conexiones o piezas de equipo físico (dispositivos) como un módem. Los objetos tiene atributos, que pueden ser manipulados, tales como contadores y estados. En lo que a la administración de sistemas concierne, el objeto es una abstracción de un recurso mismo.

El objeto es entonces la información acerca del elemento en cuestión, incluyendo los datos y las operaciones que se pueden realizar sobre estos datos. El administrador puede invocar funciones de modo similar a la llamada a una subrutina, con los parámetros de la llamada. Las acciones resultantes son las que han sido formalmente dadas en la definición de objeto.

Cada objeto administrado es un ejemplo de una clase de objetos. Cada clase de objetos manejada es un nombre establecido de objetos manejados que comparten la misma definición general y tienen el mismo conjunto de atributos. Los objetos en OSI, además, pueden relacionarse en jerarquías de dos tipos, un tipo es los subordinados dentro de uno superior (ej. MAC dentro de la capa de enlace). El otro tipo es una subclase dentro de una clase (ej. El módem es una subclase de la clase de convertidores de señal). Una subclase hereda todas las propiedades de la clase superior, de la cual descende.

Como parte de la definición de objetos encontramos:

- *Las operaciones de administración* que pueden ser ejecutadas sobre el objeto
- *El efecto de estas operaciones* sobre el objeto y sus atributos
- *Las notificaciones* que el objeto puede emitir
- *Las condiciones* que causan la notificación

CAPITULO VI

APLICACIONES

CAPITULO VI: APLICACIONES

Para el desarrollo de aplicaciones Cliente/Servidor se pueden emplear un rango de herramientas que van desde el uso de lenguajes de tercera generación hasta ambientes de desarrollo, no siendo exclusivo de algún tipo de herramienta, el desarrollo de aplicaciones Cliente/Servidor. Las principales aplicaciones de esta arquitectura están tomando forma principalmente en:

- Base de datos SQL
- Monitores TP
- Groupware
- Objetos Distribuidos

6.1 BASE DE DATOS SQL

La tendencia más importante entre servidores de bases de datos de cualquier tipo es el surgimiento de SQL como la *lengua franca* para la manipulación definición y control de datos, SQL es un poderoso conjunto orientado a lenguaje que consiste de unos pocos comandos.

6.1.1 FUNCIONES DE SQL

El Lenguaje SQL es usado para ejecutar operaciones de datos complejas con algunos comandos simples en situaciones que requerirían cientos de líneas de código convencional. A continuación se presenta una lista parcial de funciones:

- SQL es un lenguaje de consulta interactivo para consultas de acuerdo a necesidades específicas. SQL fue diseñado originalmente como un lenguaje de consulta para el usuario final. Sin embargo, los modernos front-ends y los gráficos para bases de datos, SQL son mucho más intuitivos para usar.
- SQL es un lenguaje de programación de bases de datos. Este puede unirse en lenguajes como C, C++ y COBOL para acceder datos o puede ser llamado usando el conjunto de interface X/Open. SQL ofrece un lenguaje consistente para programar con datos. Esto aumenta la productividad del programador y ayuda a producir sistemas más fáciles de mantener y cambiar.
- SQL es un lenguaje de definición y administración de datos. El lenguaje de definición de datos es usado para definir simples tablas, objetos complejos, índices, vistas, constantes de integridad referencial, seguridad y control de acceso. Todos los objetos definidos con SQL son automáticamente rastreados (y mantenidos) en un diccionario de datos activo (que es un sistema de catálogos). La estructura y organización de una base de datos SQL es almacenado en la base de datos misma.
- SQL es el lenguaje de servidores de bases de datos en red y esta siendo usado como un lenguaje universal para acceder y manipular todos los tipos de datos.

- SQL ayuda a proteger los datos en un ambiente de red multiusuario. Hace esto ofreciendo características de confiabilidad tal como la validación de datos, integridad referencial, de hacer entrega (rollback), bloqueo automático y detección (deadlock) y resolución en un ambiente LAN multiusuario. SQL proporciona un número de ventajas a desarrolladores de sistemas porque el mismo lenguaje que es usado para definir la base de datos es también usado para manipularlos.

6.1.2. SQL EN RED.

Las aplicaciones cliente/servidor son construidas frecuentemente sin usar RPC's en forma directa. La razón para esto es que los servidores son más usados para almacenar datos compartidos. Comprando un paquete integral de base de datos para manejar estos datos se puede usar las capacidades de SQL en red hechas por el fabricante para construir aplicaciones cliente/servidor. Esencialmente esto significa que se adquiere todo el código del servidor al igual que el código del cliente para los niveles de comunicaciones

La aplicación del cliente sólo necesita las operaciones de un paquete SQL e invocaría usando el API SQL proporcionado con la base de datos. Las rutinas API SQL, proporcionadas en forma integral, cuidan los detalles de usar RPC's para enviar el SQL al motor de la base de datos y regresar los resultados. Esto es una aplicación cliente/servidor completa.

API SQL.

La gran variedad de bases de datos relacionales proporcionan diferentes tipos de API's SQL. La mayoría ofrecen un conjunto de comandos SQL; sin embargo, para propósitos de cliente/servidor se enfocara a las interfaces programáticas.

El tipo más común de API SQL es el SQL integrado (embedded). El programador simplemente incluye los enunciados SQL en un programa junto con los enunciados de algún otro lenguaje de programación. Un preprocesador que viene junto con la base de datos, es usado para rastrear el código antes de la compilación así que los enunciados SQL pueden ser turnados dentro de las librerías de las bases de datos. Una vez que el preprocesador ha sustituido las llamadas apropiadas a las librerías, el programa puede ser compilado normalmente y enlazado a estas librerías. Los enunciados SQL deben de ser marcados de alguna forma para que el procesador diga cuales enunciados serán convertidos. Por ejemplo, en Oracle todos los enunciados SQL van precedidos con las palabras clave EXEC SQL para indicárselas al preprocesador. Por ejemplo,

```
.... some C language code....
```

```
EXEC SQL SELECT Name FROM Account WHERE Balance < 0;
```

```
.... more C language code....
```

Otro tipo de API es la librería de interface SQL usada por el preprocesador en la traducción de enunciados integrados. Este interface, conocida como la interface de funciones invocable, puede usualmente ser

llamada directamente por el programa de aplicación. Esto requiere un poco más de programación, ya que esta es una interfase de bajo nivel, pero frecuentemente ofrece mayor flexibilidad que al usar el preprocesador. Además hay situaciones ocasionales donde el preprocesador no puede ser usado, ya que puede crear conflicto con otras herramientas de software de desarrollo.

Algunas bases de datos (ej. DEC's Rdb) ofrecen otro tipo de API que permite módulos o líneas de código SQL definidos por el usuario para ser llamados por un programa de aplicación.

Las líneas de código son escritas en un estilo de lenguaje SQL con una sintaxis única que establece el fabricante de la base de datos. Algunos lenguajes que permiten estas líneas de código ofrecen control de flujo (ej. loops y enunciados if), y todos proporcionan un medio de paso de parámetros de y hasta las líneas de código al programa de aplicación. Las líneas de código son normalmente almacenadas en archivos separados del programa de aplicación y son compiladas usando un modulo compilador para este propósito especial suministrado por el fabricante. El programa de aplicación es escrito en un lenguaje de programación tradicional (ej. C o COBOL), compilado normalmente y enlazado entonces a las líneas de código SQL ya compiladas. Para el enlazador las líneas de código SQL son simplemente procedimientos externos que el programa de aplicación llama.

El SQL invocado via un API puede ser estática o dinámica. Los enunciados SQL estáticos son especificados completamente cuando el programa es escrito (ej. Sybase) y no puede cambiar en tiempo de corrida. La ventaja del SQL estático es que la base de datos puede determinar potencialmente una estrategia de búsqueda para consultas (queries) en tiempo de compilación. Cuando el SQL estático es invocado por el programa, el motor de la base de datos no tiene el gasto de la optimización de la consulta y puede ejecutar el SQL mas rápido. No todas las bases de datos hacen esto.

Los enunciados SQL dinámicos son construidos sobre el programa cuando este corre, usando extensiones SQL propietarias (tales como declaraciones PREPARE y EXECUTE en Oracle).

Las consultas SQL dinámicas (si son optimizadas del todo) deben ser optimizadas en tiempo de corrida por el motor de bases de datos. Esto introduce un gasto extra, pero también proporciona gran flexibilidad en la clase de programas que pueden usar esta API.

Ya que las bases de datos relacionales son diseñadas para operar en tablas de datos simultáneamente, hay una impedancia de tipos mal relacionados, con la mayoría de los lenguajes de programación que están diseñados para trabajar con variables de datos en una sola vez, por ejemplo, el resultado de unir una base de datos es frecuentemente varios renglones de datos, los cuales un programa los tiene que pasar uno a la vez. Para sobreponerse de estos tipos mal relacionados, la mayoría de los API's de bases de datos suministran uno o más cursores que pueden ser manipulados por el programa de aplicación. Un cursor es esencialmente un puntero dentro de los renglones de datos recuperados, marcando la actual posición de el programa en los datos en algún momento.

Algunos errores encontrados durante la invocación de un API SQL deben ser comunicados al programa de aplicación de alguna forma. Algunos API's ofrecen un código de regreso en cada interface llamada, mientras que otros usan una área global de datos predefinida (llamada frecuentemente Area de Comunicación SQL) para reportar errores, avisos y otra información del estatus.

6.1.3 SERVIDOR DE BASE DE DATOS.

En una aplicación cliente/servidor centrada en la base de datos, el cliente usualmente solicita datos y datos relacionados con servicios (como ordenamientos y filtros) de un servidor de base de datos. El servidor de base de datos, también conocido como el motor de la base de datos, responde a las solicitudes de los clientes y proporciona acceso seguro a datos compartidos. Una aplicación cliente puede con una sola instrucción, recuperar y modificar un conjunto de registros del servidor de base de datos. El motor de base de datos SQL puede filtrar los resultados de la consulta estableciendo los resultados con un considerable ahorro en comunicación de datos.

Un servidor SQL maneja el control y ejecución de comandos SQL. Proporciona la visión lógica y física de los datos y genera planes de acceso optimizados para ejecutar los comandos SQL. En adición, la mayoría de servidores de base de datos ofrecen características de administración y utilerías que ayudan a manejar los datos. Un servidor de base de datos también mantiene tablas de catalogo dinámicas que contienen información acerca de los objetos SQL almacenados dentro de el.

Ya que un servidor SQL permite a múltiples aplicaciones acceder a la misma base de datos al mismo tiempo, debe proporcionar un ambiente que proteja la base de datos contra una variedad de amenazas tanto internas como externas. El servidor maneja la recuperación, concurrencia, seguridad y aspectos de consistencia de una base de datos. Esto incluye controlar la ejecución de una transacción y deshacer sus efectos si falla.

6.1.4. ARQUITECTURAS DE BASES DE DATOS

Las figuras 6.1, 6.2 Y 6.3 muestran tres arquitecturas que las bases de datos usan para manejar en forma remota bases de datos cliente: proceso por cliente (process-per-client), multihilos e híbrido.

A continuación se presentan las características de estos tres métodos:

- La arquitectura de *proceso por cliente* proporciona un máximo de protección de los procesos dando a cada base de datos cliente su propio espacio de dirección para procesos. La base de datos corre en uno o más procesos formados por separado. La ventaja de esta arquitectura es que protege a un usuarios de los otros y también protege al administrador de la base de datos de los usuarios. Además, el proceso puede ser fácilmente cambiado a otro procesador o una máquina multiprocesador SMP. Ya que la arquitectura depende del sistema operativo local para los servicios de multitarea, la desventaja de proceso por cliente es que utilizan más memoria y recursos del CPU que los esquemas alternativos. Puede ser lento ya que los procesos se intercambian y usan IPC. Sin embargo, estos problemas pueden ser superados fácilmente

con el uso de monitor de procesamiento de transacciones (TP monitor) que maneje una ronda de procesos reusables.

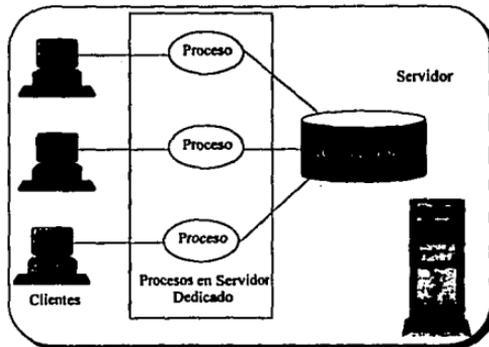


Figura 6.1 Arquitectura de Proceso por Cliente

La arquitectura *multihilos* ofrece la mejor ejecución corriendo todos los usuarios conectados, las aplicaciones y la base de datos en el mismo espacio de direcciones. Esta arquitectura proporciona su propio programador de tiempos interno y no depende del sistema operativo local y ni de sus esquemas de protección de direcciones. La ventaja es que conserva memoria y capacidad de trabajo del CPU pues no requiere de intercambios frecuentes de contexto.

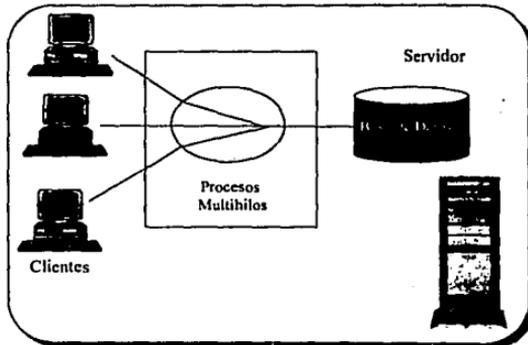


Figura 6.2 Arquitectura Multihilos

Además, las implementaciones en el servidor tienden a ser más portables entre diferentes plataformas ya que no requiere de muchos servicios del SO local. La desventaja es que un mal comportamiento de las

aplicaciones del usuario puede tirar todo el servidor de la base de datos y sus tareas. Además, los programas de usuario que tienen una duración larga puede consumir todos los recursos del servidor. Finalmente la programación previa de tareas ofrecida por el servidor tiende a ser inferior que la del SO.

- La arquitectura *híbrida* consiste de tres componentes: 1) red multihilos atenta a escuchar que participe en la conexión inicial de tarea al asignar el cliente al despachador; 2) despachadores son tareas que ponen mensajes en una cola de mensajes internos y entonces toma la respuesta y la envía de regreso al cliente; y 3) El servidor de procesos reusables y compartidos que remueve los trabajos para liberarlos de la cola, los ejecuta y coloca la respuesta sobre una cola de salida. La ventaja de esta arquitectura es que ofrece un ambiente protegido para correr las tareas de usuario sin asignar un proceso permanente a cada usuario. Las desventajas son colas latentes. Esta arquitectura es en apariencia buena, su carga de balance no es tan buena como en un monitot TP. De hecho, las colas pueden estar en forma de monitores TP teniendo sus propios algoritmos de programación de tareas. El primer servidor de base de datos que implementa esta arquitectura es Oracle7.

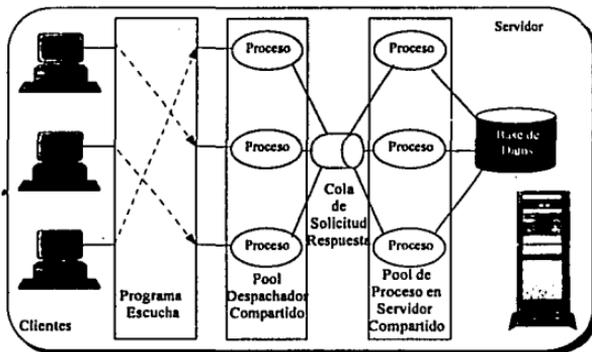


Figura 6.3 Arquitectura Híbrida

6.1.5 INTEROPERABILIDAD

Es importante remarcar que SQL es un estándar para operaciones de base de datos relacionales, no un protocolo de comunicaciones, no hay nada inherente en SQL que lo haga especialmente conveniente para comunicaciones cliente/servidor. Su popularidad es debido principalmente a su poder expreso para manipular bases de datos relacionales. SQL es un lenguaje declarativo, por el cual un usuario o aplicación puede especificar que datos quieren. No le dicta al motor de base de datos como recuperar los datos o tampoco contiene algún detalle acerca de como navegar a través de la base de datos. Esto da como resultado una

separación natural de responsabilidades entre el cliente, el cual determina que datos son de interés, y el servidor, el cual determina como manejarlos. Una importante implicación de estos es que los detalles de como el servidor tiene organizados los datos, donde esta oculto al cliente. Ya que el cliente usa SQL, él no tiene que saber como el orden en que los datos son organizados para accederlos.

SQL ha llegado a ser la lengua franca para acceder las bases de datos relacionales; el mismo comando SQL estandarizado puede ser usado virtualmente con las principales bases de datos relacionales. Ya que SQL es ampliamente soportado, los motores de bases de datos de diferentes proveedores parecen identicas a los usuarios.

El estandar ANSI para SQL (ANSI X3.135.1) define dos niveles de aceptados de conformidad, el nivel 1 consiste en los comandos del Lenguaje de Definición de Datos (Data Definition Language DDL) y Lenguaje de Manipulación de Datos (Data Manipulation Language DML). DDL incluye comandos tales como CREATE y ALTER, los cuales definen tablas y otros objetos en la base de datos. En adición a DDL también se proporciona comandos como PRIMARY KEY y FOREIGN KEY, los cuales son usados para declarar las reglas usadas en la integridad referencial. DML incluye comandos tales como SELECT, INSERT y DELETE, los cuales operan sobre renglones y columnas dentro de las tablas. El nivel 1 además define un cursor capaz de procesar un programa renglon por renglon. El nivel 2 consiste del nivel 1 más los comandos del Lenguaje de Control de Datos (Data Control Language DCL). DCL incluye comandos tales como COMMIT, ROLLBACK Y GRANT, los cuales controlan la seguridad, integridad, concurrencia y recuperación de la base de datos.

Desafortunadamente, hay muchos aspectos de los API's SQL que no cubre el estandar ANSI.

ASPECTOS DE EJECUCION

Usar SQL en la red no siempre es eficiente. Si la operación SQL selecciona una gran cantidad de datos para ser manipulados por el cliente, puede ocurrir una congestión de la red. Por ejemplo supongase que una transacción une dos tablas, dando como resultado una recuperación de 100,000 registros. Si la aplicación necesita simplemente computar algunas operaciones basandose sólo en algunos registros o verificar que alguna regla no sea violada, no hay necesidad de que estos registros sean enviados a través de la red. En lugar de enviar varios enunciados SQL individuales, tiene más sentido agruparlos como una operación para ser ejecutada en el servidor. Cuando un cliente necesite ejecutar la operación, debiera indicarle al servidor que debe invocarla. Cuando la operación se completa, un estatus completo y los resultados pueden ser regresados al cliente. Este metodo involucra mucho menor trafico en la red que si la operación fuera hecha en el cliente.

Algunas bases de datos, como Sybase e Ingres, han agregado extensiones a su SQL de red para permitir a los clientes invocar procedimientos almacenados remotos. Un procedimiento almacenado es una subrutina proporcionada por el usuario que corre sobre el servidor de base de datos. En lugar de enviar varios comandos SQL separados (y recibir sus resultados en los intermedios), el cliente envia un sólo mensaje para invocar un procedimiento almacenado. Los procedimientos almacenados son similares a los procedimientos remotos,

excepto que están almacenados en la base de datos con los datos y son accesibles sólo dentro del ambiente de la base de datos. Usualmente, pueden pasarse y devolverse parámetros de los procedimientos almacenados. Algunas bases de datos permiten definir variables e incluir lógica del flujo de control dentro de un procedimiento almacenado. Otras sólo permiten enunciados SQL en un procedimiento almacenado.

Por ejemplo, los procedimientos almacenados en Sybase permiten variables y desviación en la lógica de control que es usada con los enunciados SQL almacenados en el servidor de base de datos. Cuando son almacenados, la base de datos distribuye los comandos, chequea la sintaxis y formula un plan de ejecución que se usará. Cuando son invocados, Sybase ejecuta el SQL precompilado mucho más rápido que un comando SQL normal.

La figura 6.4 ilustra cómo los procedimientos almacenados pueden ser usados para reducir el tráfico en la red. El uso de un procedimiento almacenado es transparente a otras consultas de base de datos a menos que el procedimiento almacenado use una gran cantidad de recursos del servidor. Los procedimientos almacenados son típicamente usados para forzar la integridad de datos o reglas del negocio. Además, ya que un procedimiento almacenado corre mucho más rápido que una transacción SQL no compilada, frecuentemente las transacciones invocadas son candidatos potenciales para procedimientos almacenados.

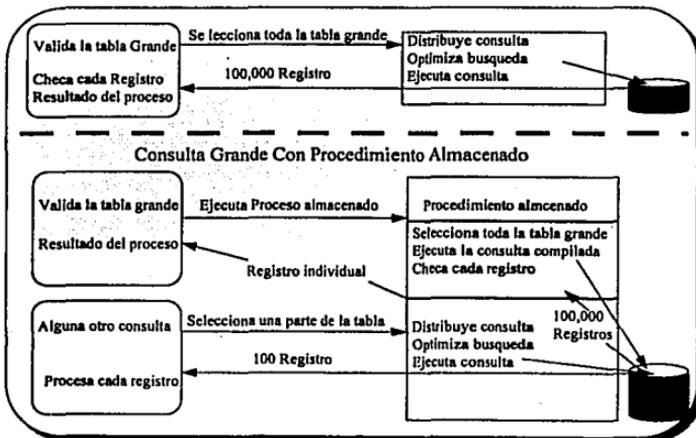


Figura 6.4 Consultas con procedimientos almacenados

Usualmente, los procedimientos almacenados son invocados explícitamente por enunciados SQL de red especiales. Sin embargo, algunas bases de datos permiten que sean iniciados implícitamente al cambiar los datos. Un disparador (trigger) es un tipo especial de procedimiento almacenado que es automáticamente

invocado por el motor de la base de datos siempre que ocurra un evento específico en la base de datos. Por ejemplo, un disparador puede activarse basándose en un campo específico que está actualizándose en una tabla. El disparador puede asegurarse que otros campos se mantengan consistentes o provocar que una transacción esté ejecutándose.

El uso de disparadores y procedimientos almacenados debe ser balanceado con el gasto extra que producen en el servidor. Recordando que los recursos de procesamiento del servidor están compartidos por todos los clientes. Si una operación es computacionalmente cara, es usualmente mejor que esta use el poder de procesamiento del cliente dedicado a ella. En forma alternativa, si una operación hace uso intensivo de la E/S, es un candidato potencial para un procedimiento almacenado para evitar la ejecución de E/S a través de la red.

La optimización de consultas y el cuidado en un nivel muy fino de la base de datos (nivel registros, etc.) son características esenciales de una base de datos que ofrece SQL de red. Ya que el servidor de base de datos es un recurso compartido, el procesamiento ineficiente de consultas puede afectar mucho al cliente. Por ejemplo, suponiendo que un cliente envía una solicitud SQL que está pobremente formada y resulta en una ineficiente tabla de consulta. Esto podría suceder fácilmente uniendo una gran tabla a dos pequeñas tablas en vez de hacerlo en el orden inverso. Todas las transacciones SQL subsecuentes sufrirán del decremento en las acciones del servidor hasta que la unión ineficiente de las tablas se complete. Sin embargo, si la base de datos tiene una optimización estática de consulta, el costo de la consulta pobremente formada será comprada con estrategias alternativas y la consulta será reestructurada antes de ser aplicada.

Idénticamente, las bases de datos distribuidas deben incorporar optimización de consultas en la red. Por ejemplo, supóngase que una tabla de un millón de registros es mantenida en un servidor y varias tablas pequeñas en otro. Si el cliente trata de unir la tabla con un millón de registros a varias tablas con 10 registros, puede resultar en un considerable tráfico en la red. En forma alternativa, si uno primero las tablas pequeñas y luego se aplica a la tabla con un millón de registros, habrá un menor tráfico en la red. Esta optimización debe realizarse en el cliente, ya que los datos serán ensamblados de múltiples servidores. Cada servidor puede tener una buena optimización de información con sus propios datos, pero el cliente es el único lugar donde toda la información de la consulta se conocerá.

Las bases de datos distribuidas también requieren extensiones SQL para soportar actualizaciones distribuidas. Típicamente, un protocolo de entrega en dos fases, es implementado en estos casos, mostrado en la figura 6.5. Durante la primera fase, una transacción en el manejador de entrega envía la actualización distribuida a cada servidor de base de datos participante. Cada servidor de base de datos adquiere la llave necesitada para aplicar la actualización. Cuando están listos, envían un mensaje de confirmación al manejador de entrega.

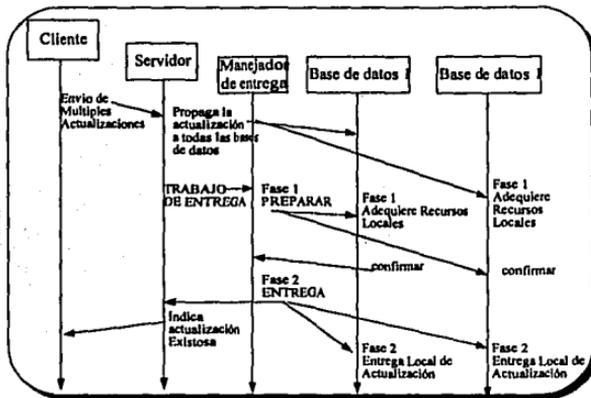


Figura 6.5 Protocolo de entrega de dos fases

Cuando el manejador de entrega recibe la confirmación de todos los participantes, entra la segunda fase de entrega notificando al servidor de base de datos la entrega de la transacción. Si no se logra la confirmación de cada servidor, el manejador de entrega cancela la transacción y todos los servidores quitan sus llaves. Una vez que se entra en la segunda fase, cada servidor debe garantizar la entrega de la transacción. Si hay una caída, El servidor de base de datos checará al reestablecerse que estaba haciendo y detectará que había enviado una confirmación al manejador de entrega. Entonces el manejador de entrega decidirá si se cancela la transacción o se reintenta.

Hay muchas cuestiones asociadas con las bases de datos distribuidas; sin embargo no es nuestra intención profundizar en estos tópicos. Para una revisión más completa se puede consultar la bibliografía al final del capítulo.

6.2. MONITORES TP

No es posible crear aplicaciones de misión crítica sin administrar los programas (o procesos) que operan en la información. Este es el motivo por el cual, en el mundo de las macrocomputadoras, un monitor TP viene integrado con cada una de las bases de datos de misión crítica. Los monitores TP administran los procesos y orquestan los programas dividiendo las aplicaciones complejas en varias piezas de código llamadas transacciones. No es descabellado asumir que cada máquina conectada a la red tendrá un monitor TP que la representará en las transacciones globales.

Las transacciones son más que simples eventos comerciales; se han convertido en una filosofía de diseño de una aplicación que garantiza la fuerza en sistemas distribuidos. Bajo control de un monitor TP, una

transacción puede ser administrada desde su punto de origen (por lo general una estación cliente) a través de uno o varios servidores y regresar a la estación cliente de origen. Al terminar una transacción, todas las partes implicadas deciden si tuvo éxito o no.

La transacción es el contrato que conecta a la estación cliente con uno o más servidores. Es la unidad fundamental de recuperación, consistencia y concurrencia en un sistema cliente/servidor. Naturalmente, todos los programas participantes deben adherirse a la disciplina transaccional; de lo contrario, un sólo programa defectuoso puede corromper todo un sistema. En un mundo ideal, todos los programas cliente/servidor serán escritos en forma de transacciones.

Los modelos de transacción definen cuándo comienza una transacción, cuándo termina y cuáles serán las unidades de recuperación en caso de que haya una falla. El modelo de transacción plana, ha sido el caballo de batalla de la generación actual de monitores TP (y otros sistemas transaccionales). En una transacción plana, todo el trabajo que se realiza dentro de sus límites, ocurre al mismo nivel. La transacción comienza con un `begin_transaction` y termina ya sea con un `commit_transaction` o un `abort_transaction`. Es todo o nada, no hay forma de cometer o abortar partes de una transacción plana. De cualquier forma, nuevos modelos de transacciones ofrecen un mejor control sobre los hilos de una transacción y pueden asimilar más de cerca a sus contrapartes del mundo real.

La mayoría de las alternativas de las transacciones planas, extienden el flujo del control más allá de la unidad simple de trabajo, ya sea cambiando unidades de trabajo en secuencias lineales de minitransacciones o a través de transacciones secundarias anidadas. Los efectos de una transacción secundaria se vuelven permanentes después de que se envía un cometido local y el cometido de todos sus ancestros. Si una transacción madre aborta, todas las transacciones descendientes abortan, ya sea que envíen cometidos locales o no. Lo bueno de esto, es que las transacciones secundarias pueden ser ejecutadas en nodos distintos.

Los monitores TP fueron inventados para ejecutar aplicaciones que sirvan a miles de estaciones cliente. Interponiéndose entre estaciones cliente y servidores, los monitores TP pueden administrar las transacciones, dirigirlos a través de los sistemas, nivelar su ejecución y reiniciarlos tras un fallo. Un monitor TP puede administrar recursos transaccionales en un sólo servidor o a través de servidores múltiples, además de que puede cooperar con otros monitores TP en arreglos federados. Los monitores TP también pueden llevar a cabo un gran acto de encauce que ayuda al sistema operativo y a los administradores de recursos de servidor a manejar grandes cantidades de estaciones cliente.

X/Open y el Grupo de Administración de Objetos (OMG, Object Management Group) han creado normas complementarias que definen la forma en que los monitores TP interactúan con las aplicaciones, con los administradores de recursos y con otros monitores TP tanto en entornos procesales como de objetos distribuidos. Algunas buenas herramientas están disponibles para ayudarle a crear aplicaciones de monitor TP.

Los monitores TP probablemente tienen demasiada capacidad como para una aplicación departamental de un sólo fabricante/un sólo servidor. Esta es la razón por la cual han tardado tanto en conseguir el éxito. Además, los distribuidores aún no han podido llegar al nivel del mercado del software compacto, y no han podido explicar las ventajas que los monitores TP pueden ofrecer.

6.3 GROUPWARE

El groupware consta de cinco tecnologías de fundación unidas, para dar soporte al trabajo colaborativo: administración de documentos multimedia, flujo de trabajo, correo electrónico, conferencias y programación de agenda. El groupware es otra tecnología de macrocomputadora minimizada.

Es un modelo genuinamente nuevo para la computación cliente/servidor. Ayuda a los usuarios a juntar datos no estructurados, incluyendo texto, imágenes, faxes, correo y procedimientos de conferencias en línea, y a organizar esa información en forma de un conjunto de documentos. Además, permite a los usuarios visualizar esos documentos, almacenarlos, replicarlos y dirigirlos en cualquier parte de la red. El documento multimedia es para el groupware, lo que una tabla para una base de datos SQL: la unidad básica de administración.

El groupware es óptimo para el arte de la administración de bases de datos de documentos y hace efectivo el uso del correo electrónico, una forma preferencial de software medio. El correo electrónico es para los procedimientos electrónicos una de las formas más sencillas de comunicarse con los humanos. Asíncrono por naturaleza, es ideal para la forma en que los negocios funcionan realmente. El correo electrónico se encuentra en todos lados, con más de cincuenta millones de apartados postales electrónicos globalmente interconectados.

Administrar procesos de datos por medio de flujo de trabajo, es otro aspecto revolucionario del groupware. En un flujo de trabajo, la información (y en ocasiones las funciones) pasa de un programa al siguiente en ambientes cliente/servidor estructurados o no estructurados. El software de flujo de trabajo actual simula en forma electrónica la actividad colaborativa del mundo real. El trabajo puede ser dirigido en formas que corresponden a las comunicaciones entre oficinas. Se pueden crear rutas secuenciales, rutas paralelas (por ejemplo, vías de acceso alternas), rutas con ciclos de retroalimentación, rutas circulares, etc.

Un buen paquete de flujo de trabajo le permite especificar los criterios de aceptación de una etapa a otra. Así que, el flujo de trabajo lleva la información a las personas (y a los programas) que pueden actuar con ella. También puede coordinar el software existente y rastrear procesos para asegurarse de que el trabajo quede realizado por las personas adecuadas.

Groupware proporciona muchos de los componentes que requerimos para crear aplicaciones cliente/servidor. La tecnología también comienza a usurpar el campo de sus competidores. Por ejemplo, Lotus Notes, vía DataLens, puede acceder a información almacenada en bases de datos SQL; además, permite que las aplicaciones SQL tengan acceso a la información de Notes por medio de la Conectividad Abierta de Bases de Datos (ODBC, Open Database Connectivity).

Las nuevas herramientas, integran instalaciones cliente para la formación de GUI, con datos a los que se puede acceder desde documentos o bases de datos SQL. En su mejor punto, el groupware puede combinar en forma flexible distintas tecnologías de cliente/servidor y adaptarse a la forma en que las personas hacen negocios, tanto en configuraciones adecuadas como estructuradas.

6.4 OBJETOS DISTRIBUIDOS

La tecnología de objetos distribuidos promete la mayor flexibilidad en sistemas cliente/servidor. Esto se debe a que encapsula datos y lógica comercial en objetos que pueden navegar en cualquier sitio de la red, ejecutarse en plataformas distintas, comunicarse con aplicaciones de cesión por medio de empaquetadores de objetos y administrarse a sí mismos y a los recursos que controlan.

Cuando se trata de normas, la tecnología de objetos distribuidos está más adelantada que los demás enfoques cliente/servidor. Desde 1989, un consorcio de distribuidores de objetos, llamado OMG, ha venido especificando en forma constante la arquitectura de un bus de software abierto en el cual los componentes de objetos creados por distintos fabricantes puedan interoperar a través de redes y de sistemas operativos. Actualmente las 440 compañías son miembros de la OMG, y el bus de objetos está en camino de convertirse en la madre de todo el software medio de cliente/servidor.

El secreto del éxito de la OMG reside en que definió la forma de especificar una interfaz entre un componente y el bus de objetos, utilizando tecnologías que se encontraban funcionando como un modelo, aunque no prescribió la forma de implementar esas especificaciones. Las especificaciones están escritas en Lenguaje de Definición de Interfaz (IDL, Interface Definition Language), independiente de cualquier lenguaje de programación. Los componentes especifican en IDL los tipos de servicios que proporcionan, incluyendo los métodos que exportan y sus parámetros, atributos, administradores de errores y relaciones hereditarias con otros componentes.

IDL se convierte en el contrato que une a las estaciones cliente con los componentes del servidor. Lo bello de IDL es que puede ser utilizado fácilmente para encapsular aplicaciones existentes. No es necesario escribir nuevamente el inventario completo de aplicaciones para tomar ventaja de la tecnología de objetos distribuidos.

El bus de objetos proporciona un ORB que permite a las estaciones cliente invocar métodos en objetos remotos ya sea en forma estática o dinámica. Si una interfaz componente ya está definida, se puede vincular el programa con un cabo generado por IDL para invocar sus métodos. De lo contrario, se puede descubrir la forma en que la interfaz trabaja al tiempo de la ejecución, consultando un depósito interfaces especificado por la OMG.

A finales de 1994, la OMG aprobó una serie de especificaciones llamada Arquitectura de Corretaje de Requisiciones de Objetos Comunes (CORBA 2.0, Common Object Request Broker Architecture), que define las

bases de ORB a ORB basadas en TCP/IP. CORBA 2.0 define así mismo un servicio opcional de comunicaciones interORB basado en DCE. La nueva especificación del depósito de interfaces, define las extensiones que permiten a los componentes generar identificadores (ID) globales universales para sus interfaces (utilizando IDL), para asegurar que sean únicos a nivel intergaláctico.

Además de definir el bus de objetos, la OMG ha especificado una serie extensa de servicios basados en ORB para crear y eliminar objetos, acceder a ellos por nombre, almacenarlos, extemar sus estados y definir relaciones complejas entre ellos. A finales de 1994, la OMG definió también una serie comprensible de servicios de objetos transaccionales. La idea es que el usuario pueda crear un objeto ordinario y hacerlo transaccional, bloqueable y persistente, haciendo que herede los servicios apropiados utilizando registros IDL sencillos.

La OMG creó alianzas importantes para asegurarse de que sus normas sean universalmente aceptadas. El servicio de persistencia definido por CORBA está alineado muy de cerca con el nuevo Grupo de Administración de Bases de Datos de Objetos (ODMG-93, Object Database Management Group) para las bases de datos de objetos. Los servicios de transacción de objetos pueden interoperar con transacciones procesales definidas por X/Open. La OMG está trabajando también con X/Open para ayudar a definir interfaces de administración de sistemas basadas en ORB, incluyendo la seguridad.

Además, CI Labs, el consorcio de las compañías responsables de OpenDoc, seleccionó a CORBA como su modelo de objetos para las comunicaciones entre componentes. Tanto Taligent como OpenStep proporcionan puertas de acceso CORBA para sus comunicaciones externas de objetos. Incluso Microsoft se acercó a la OMG a finales de 1994 con una propuesta para implementar una puerta de acceso oficial de OLE a CORBA.

Parece que la comunidad de objetos está en camino para desarrollar una infraestructura de objetos que pueda satisfacer la demanda de la era cliente/servidor. Los objetos distribuidos con el paquete de componentes y la infraestructura apropiados, pueden proporcionar los bloques más novedosos de desarrollo para crear soluciones cliente/servidor, incluyendo paquetes integrados de objetos comerciales cooperativos.

6.4 DESARROLLO DE SECOFI

Al igual que todo en la vida, la informática y los sistemas de cómputo son una especialidad que evoluciona constantemente, incluso se puede decir que es una de las ramas de la tecnología que con mayor frecuencia presenta cambios, tanto en los equipos, programas y técnicas que lo conforman, como en las diferentes filosofías de trabajo que le dan sentido.

Así desde la fabricación de la primera computadora, la tecnología ha avanzado con gran rapidez en lo que al tamaño, capacidad y potencial de los equipos se refiere, así como en el tipo y forma de uso de los programas que en ellos se ejecutan. De los antiguos, grandes y lentos "monstruos", se ha pasado a pequeños y poderosos equipos de cómputo, los cuales en un espacio menor y sin condiciones especiales de medio ambiente,

proporcionan una capacidad casi limitada de procesamiento. Igualmente, los complicados sistemas de programación que requerían de expertos altamente capacitados para su desarrollo y utilización, han sido substituidos por los ambientes gráficos, totalmente amigables e intuitivos para el usuario final.

La Secretaría de Comercio y Fomento Industrial (SECOFI), a través de la Dirección General de Planeación Informática es una de las dependencias de la Administración Pública que con mayor interés, ha atendido la evolución que ha ocurrido en esta tecnología, tratando de incorporar sus avances a la atención de sus necesidades. En el presente trabajo se redactará de manera breve, como ha venido transformándose la infraestructura de cómputo en la SECOFI, hasta llegar a una red de área amplia a nivel Nacional.

ANTECEDENTES

Los inicios del ambiente de cómputo en la Secretaría de Comercio y Fomento Industrial tuvieron origen, con la compra y adquisición de equipo de cómputo a nivel Mainframe, ya que los principales procesos a automatizar eran los sistemas de las áreas administrativas, que manipulaban grandes cantidades de información, y que en ese entonces no podían ser soportados en equipos de microcomputación dado que la tecnología a pesar de tener ventajas sobre el esquema adoptado, era muy nueva en el país. Por lo tanto, se instaló un equipo Sperry Univac 1160 en donde se desarrollaron sistemas para procesar algunos sistemas administrativos y otros relativos a la gestión de la Secretaría. Además se brindaba apoyo de cómputo a otras entidades coordinadas.

Para el año de 1985, y debido a cambios en los esquemas de trabajo de la función informática, se adquirió un equipo de tipo "Mainframe" Burroughs modelo A9-D con la finalidad de automatizar algunos procesos adicionales de gran importancia para la Secretaría. La A9, como se le conoce, es una computadora con orientación al teleproceso y con software especializado para diversas tareas, como son el manejo de grandes bases de datos, la generación de reportes y pantallas en los manejos de directorios de datos.

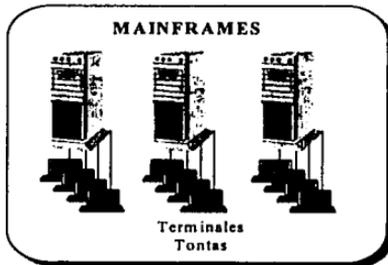


Figura 6.6 Esquema Original

Hacia fines del mismo año, la dependencia absorbió parte de las funciones del desaparecido Instituto Mexicano de Comercio Exterior y también de sus recursos. Lo que le trajo como beneficio temporal, la adquisición de un equipo Burroughs B-3700, cuya orientación era el procesamiento "batch" y un equipo

Burroughs B-5900. Debido al avance de la tecnología de cómputo, el primero de dichos equipos, fue desactivado en el año de 1986, y el segundo de ellos, se mantuvo en operación hasta el año de 1989, trabajando de manera conjunta con el equipo A9.

SECOFI Y LAS REDES LOCALES

En el año de 1986, a fin de impulsar la desconcentración de funciones de la Secretaría se adquirieron 70 microcomputadoras Olivetti M24, computadoras personales, con 640 KB en Memoria RAM, 20 MB en disco duro y con procesador Intel 8086, las cuales fueron distribuidos entre las Delegaciones Federales (provincia) más importantes y en las áreas normativas responsables de programas desconcentrados, a fin de poder compartir información entre ellas. Esos fueron los inicios de la Red Telemática de SECOFI, la cual empleaba entonces, como medio de transmisión de datos la red pública TELEPAC.

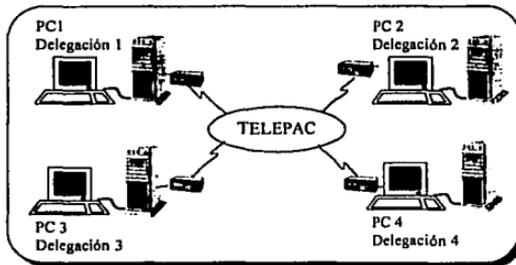


Figura 6.7. Red Telemática

En 1988, al cambio de administración y con la nueva orientación de las funciones de SECOFI, se inició una transformación radical de los servicios de cómputo, enmarcada en el Programa de Modernización Informática (PMI), cuyos puntos centrales fueron el fortalecimiento de la infraestructura de cómputo de las diversas áreas de la Secretaría así como la desconcentración de las funciones de diseño y desarrollo de sistemas.

Para lograr lo anterior y con base en un análisis de las tendencias del mercado de cómputo en el área de las comunicaciones, se realizó la adquisición de las primeras redes locales de la Secretaría basadas en Ethernet y Sistemas Operativos Novell Netware uno de los más poderosos en esos años y con equipos Olivetti Modelos M-290, con procesadores Intel 80286, Memoria RAM de 1 MB y discos duros de 20 MB, para estaciones de trabajo y para reforzar los esquemas de procesamiento se adquirieron equipos Hewlett Packard modelo RS-25, para funcionar como servidores de archivos de dichas redes. Estos equipos tienen un procesador Intel 80386 y diferentes capacidades en almacenamiento primario y secundario de acuerdo a las necesidades de las áreas, las cuales iban de 4 a 8 MB en RAM y de 300 a 600 MB en disco duro. Además de la adquisición de dichos equipos, se integraron todos los dispositivos necesarios para unir en redes locales los equipos ubicados en las Unidades Administrativas de la Dependencia.

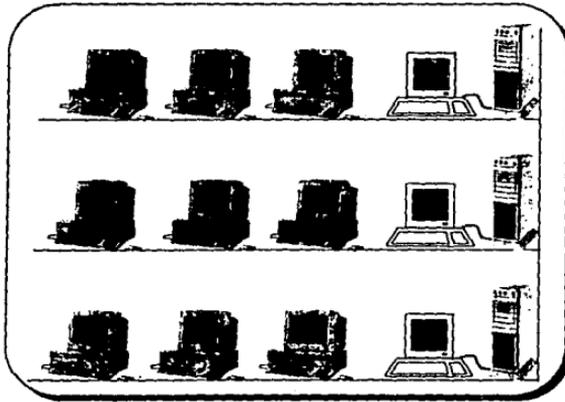


Figura 6.9 Estructura del Backbone

Durante ese año, se reforzó ampliamente la estructura de cómputo de las Delegaciones Federales, instalándose redes de cómputo en cada una de ellas, las cuales tenían como servidores de archivos equipos Olivetti M300, con procesador 80386sx con 4 MB en RAM y 100 MB en disco duro. Al mismo tiempo, en oficinas centrales, se incorporaron a las diferentes redes locales estaciones de trabajo Olivetti M300 y se adquirieron servidores de red para nuevas redes, Hewlett Packard Modelo Vectra 486, con capacidades de hasta 1.3 GB en disco duro. Además de lo anterior se estructuró la central de comunicaciones con provincia, mediante la cual se realiza actualmente el intercambio de correo electrónico entre las Delegaciones Federales y la Zona Metropolitana

En 1991 continuó incrementándose la cantidad de equipos instalados en la Dependencia con la inclusión de microcomputadoras Olivetti modelo PCS386 y servidores de archivos Olivetti LSX5020.

Respecto al Mainframe, y dado que para estos años, el auge de las redes locales y la estabilidad que éstas presentaban era bastante aceptable, se comenzó una estrategia de Downsizing, es decir, los sistemas desarrollados para la A9 se comenzaron a fabricar de manera paralela en el ambiente de redes locales, utilizando manejares de bases de datos, basados en un lenguaje estándar como lo es SQL y paquetes de desarrollo en ambiente gráfico, de tal forma que el 31 de Octubre de 1991, el equipo A9 fue oficialmente desactivado, manteniéndose hoy e día sólo en operación esporádica para respaldo de información histórica. De esta manera, la SECOFI consumó una migración de esquema informativos grandes a ambientes de trabajo más pequeños pero más poderosos.

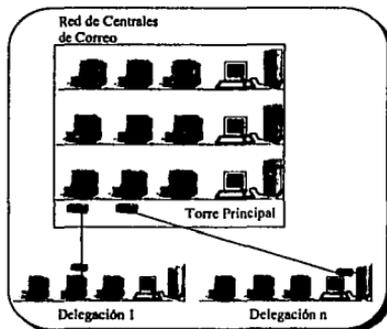


Figura 6.10 Red de Comunicaciones

A finales de ese mismo año, se instaló el primer enlace de Microondas (a 10 Mbps) entre la Torre Ejecutiva y el edificio de Agricultura 21, dando inicio la operación de la red Metropolitana de Comunicaciones de la SECOFI, mediante la cual las redes locales de la Zona Metropolitana se comunican entre sí, formando un gran backbone metropolitano.

De igual forma, se concluyó la instalación de la Red Metropolitana de Comunicaciones, la cual incluye un enlace de microondas a la Secretaría de Hacienda y Crédito Público, que integra ambientes de trabajo UNIX con ambientes de redes locales, primer enlace de su tipo en México.

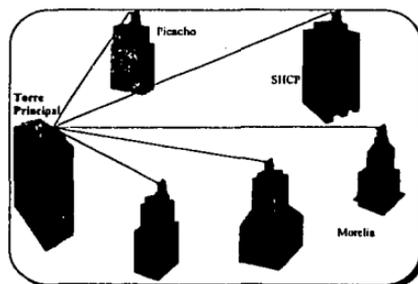


Figura 6.11 Esquema de Microondas

En el año pasado, se fortaleció la capacidad de las redes locales de las Delegaciones Federales, cambiando los servidores de archivos por equipos Olivetti PC PRO 33MT, con 4 MB en RAM y 300 MB en disco duro. Asimismo, se han instalado en la Institución estaciones de trabajo Olivetti PC PRO SX 20 y Hewlett Packard Vectra 386/20 N.

En el aspecto de software, con el objeto de facilitar el desempeño de las actividades de trabajo diarias, se han integrado ambientes gráficos de computación, los cuales favorecen la integración hombre-computadora, así como manejadores de bases de datos poderosos, para manipular la información de manera más eficiente.

Así, cabe mencionar que de igual manera que se han integrado las redes locales con las estaciones de trabajo y servidores mencionados, se han integrado una gran cantidad de dispositivos tales como impresoras de carácter, impresoras láser, digitalizadores de imágenes, ratones, unidades lectoras de CD, unidades de respaldo, unidades de cinta, etc.

LAS TENDENCIAS EN LA SECOFI

En primer término, y debido al crecimiento de las áreas de la Secretaría en aspecto de equipos, se integrarán una mayor cantidad de estaciones de trabajo, de tal manera que la actual relación de 1 equipo por cada 3 empleados, se convierta en una relación de 1 equipo por cada 2 empleados. Asimismo, se reforzará la estructura de las redes, mediante la incorporación de la tecnología de cableado estructurado y esquemas de administración global de la misma, así como la utilización de superservidores.

CONCLUSIONES:

Al gran crecimiento de las organizaciones y la mayor demanda de soluciones informáticas para realizar tareas cotidianas, en diversas áreas y a todo los niveles jerárquicos, son factores que han contribuido al desarrollo de redes de área locales (LAN).

Cuando se tiene un sistema de redes con varios usuarios operando y solicitando diversas aplicaciones al mismo tiempo es posible que la línea de comunicación se saturen y lleguen incluso a que el sistema no responda. Es por esto que se llega a la descentralización de tareas, donde cada servidor es capaz de realizar tareas específicas, evitando el desbordamiento de esta línea, así se tienen varias alternativas, en caso de que un servidor llegará a fallar, el sistema en general seguirá respondiendo.

Tomando los términos de computación distribuida, cliente/servidor y computación de red, que han constituido el nuevo modelo para redes de área local en la década de los noventa, este modelo ha sido impulsado en parte por el downsize. La introducción de supervisores y microprocesadores de alta velocidad han hecho que las soluciones de sistemas cliente/servidor sean muy atractivas y accesibles. Además de soportar nuevas aplicaciones como gráficos y videos que están basadas en normas. Esta configuración de sistema cliente/servidor es ideal para los ambientes que operan con estaciones de trabajo de alto rendimiento

Si un cliente puede ser un dispositivo conectado a un nodo de la red o un programas de software que necesita ser atendido por el servidor. Para realizar las operaciones requeridas por el modelo cliente/servidor debe existir un software especializado en ambos extremos del contexto. Por ejemplo, en un sistema de bases de datos para trabajar en red, la interfases con la que opera el usuario reside en la estación de trabajo y las funciones de almacenamiento y recuperación en el servidor. Y si es utilizado el termino cliente/servidor en aplicaciones de programas como una relación petición/suministro, se pueden tener o diseñar aplicaciones en las que un programas (cliente) pide datos a otro programa (servidor).

Los servidores deben poder manejar miles de clientes en redes complejas y grandes. Y para comunicaciones servidor a servidor, se requiere "pipe". Los superservidores potentes son convenientes en estas situaciones, al ofrecer una solución económica que considera todos los servicios que ofrecen a miles de clientes.

A diferencia de los del servidor, los requisitos de los clientes dan mas importancia a la conectividad ya que deben estar conectados a varios servidores, estén estos en un segmento local o

a través de la red. Distribuidos por toda la red, los servidores dedicados, instantáneamente soportan miles de clientes que operen a un nivel de misión crítica con servicios de aplicación, archivos, comunicación y correo. Como resultado, los datos y aplicaciones en un servidor deben de dar la impresión de ser locales a todos los clientes en cualquier parte de la red.

El nivel de rendimiento requerido por cada cliente lo determinan las aplicaciones que están siendo utilizadas. En cuanto a cableado, tal vez la mejor alternativa disponible hoy en día sea una mezcla de FDDI de 100 Mbps y Ethernet de 10 Mbps. FDDI es ideal para conectar recursos compartidos de red tales como servidores de archivo, servidores de aplicación y otros aparatos que requieren mayor ancho de banda y velocidad. Al mismo tiempo, el Ethernet de 10 Mbps maneja mejor las conexiones del cliente y el tráfico de los segmentos de la red.

Puentes y ruteadores de alto rendimiento proveen la conexión entre servidores en una red de FDDI y los clientes en los segmentos de Ethernet. Este método ofrece una solución adecuada para las distintas necesidades de ambos clientes y servidores. Al colocar los clientes y servidores en ambientes donde cada uno puede realizar su potencial, las redes alcanzarán mejores niveles de rendimiento.

Así, al terminó del presente trabajo se ha podido apreciar la importancia que tiene el trabajar en red, debido a que esto nos proporciona grandes ventajas, sobre todo el rendimiento y compartición de dispositivos o recursos. El instalar o tener una red no significa, llegar al final de un proyecto, de lo contrario es cuando nos debemos de preocupar por explotar todo lo que esta podría ofrecer. Así la mayoría de las empresas, organizadas con redes de comunicación, han permitido que los empleado trabajen en equipo y que la información tecnológica se convierta en un factor mas importante para el desarrollo y competitividad de la empresa. Todo lo anterior ha propiciado los sistemas Cliente/Servidor, con metas ambiciosas en la tecnología de la información, las cuales buscan básicamente que su desarrollo vaya a los sistemas abiertos y hacia la arquitectura Cliente/Servidor.

BLIBLIOGRAFIA

- Herbert A. Edelstein, "Database world target next generation problems", Software magazine, vol. VII, No. 6, pp 81, Mayo 1991
 - Ross E. Floyd, "An overview of FDDI: the firer distributed data interface", IEEE Journal on select areas in communications, USA, vol 7, No 7 septiembre de 1989
 - Maynarā Mass. "The Ethernet, a local area network, data link layer and, physical layer specifications, Digital Equipment Corp, Xerox Corp", Ver 2.0 Noviembre de 1982.
 - Khoshafian Setrag , Baker Brad A., Abnouz Razmik and Shepher Kevin, Intelligent offices: objet-oriented, multimedia, information management in client/server architecture. Jonh Wiley & Sons, Inc., New York, 1994
 - Jim Car. "Defining Downsizing, USA", LAN Magazine, pp 23-24, 1994
 - Bux, "A reliable token ring system for local area communications" National Telecommunications Conference, Nueva Orleans, 1981
 - H. Dineen, P.S. Leanch, N. W. Mishkin, J. N. Pato, and G. L. Wyant, "The network computing architecture and system: An environment for developing distributed applications ", IEEE computer society conference, San Francisco, February 1988.
 - Mischa Schwartz."Redes de telecomunicaciones: Protocolos, modelado y analisis" México 1992
 - Draft E. "IEEE standar 802.5 token ring access method and physical layer specifications, local area networks standars, Proyect IEEE 802" , New York, 1984
 - Dunlap Charlotte, "Cliente/Servidor firme en el mundo corporativo". México, Oct. de 1993, Personal Computing, pp 38-42,
 - Smith Patrick and BSG, "Client/Server computing", USA, SAMS, 1992.
 - Uyless Black., "Redes de computadoras: protocolos, normas e interfaces", Madrid, 1989.
 - Mc Namara John., "LAN: an introduction to the technologies", USA, 1990
 - Schatt, Stan, "Understanding LAN", USA, 1993.
 - Hernandez Becerra Martha Elena. "Los protocolos en las redes locales Ing. en compl" UNAM, México 1994
 - Cypser R. J. "Communications for cooperating systems: OSI, SNA, and TCP/IP", USA, 1992.
 - Guillermo Levine, "Introducción a la computación", México 1990
 - Taber, Dave. "Alert focal-point" Interface, Abril 1989.
-

-
- "OSI network management forum. Forum arquitectura" Issue 1, January 1990
 - Tom Sheldon, "Novell Netware 4.0 Manual de referencia", México 1994.
 - Andrew S. Tanenbaum, "Sistemas Operativos Modernos", México 1994.
 - SCO, "Lan Manager for Unix", USA 1994
 - Microsoft "Lan Manager for Dos", USA 1993.
 - Carlos Gimeno "Introducción a Novell", México 1994
 - Jose Antão Beltrão, "Redes Locales de Computadoras" México 1992
 - Addison Wesley, "Redes de telecomunicaciones: Protocolos, Modelado y Análisis, USA 1993
 - Michael Dan and Ken Netf, "Netware in the LAN", Canada 1994.
 - Jeff Woddword "El ABC de Novell Netware", Canada 1991.
 - David Janes Bazza, "Sistemas Operativos", España 1994.
 - Just in Time, "JIT, Graphical User Interface Cliente/Server", USA 1992.
 - HP Computer Systems, "Lan Manager/HP-UX for the HP 9000, USA 1994
 - HP Network, "Introduction to Data Communications", USA 1993.
-