35
2ej.

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
UNIDAD ARAGON

IMPLANTACION DE TCP/IP EN EQUIPOS DE LA FACULTAD DE QUIMICA

T E S I S

PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

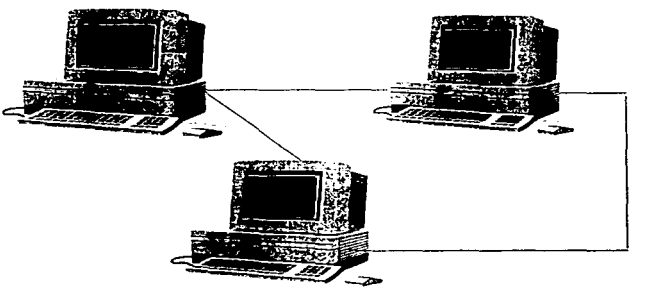
P R E S E N T A :

MARCO ANTONIO LOPEZ MELENDEZ

ASESOR DE TESIS: ING. DONACIANO JIMENEZ VAZQUEZ

SAN JUAN DE ARAGON, ESTADO DE MEXICO 1997

Implantación de TCP/IP en equipos personales



Agradecimientos

Agradezco a la vida por darme la oportunidad de culminar una meta que me he propuesto.
Agradezco a mis padres el darme la vida y esta carrera, no sería nada sin ustedes.
Agradezco a todos mis amigos por su apoyo.
Agradezco a Magda por el gran apoyo que me brindó en los momentos difíciles.
Agradezco a mi asesor sin el esta tesis no sería de un Ingeniero.
Agradezco a todos los profesores que han sembrado en mí el conocimiento.
Agradezco a esta casa de estudios permitirme entrar en sus aulas y recibir su conocimiento.

Contenido

Introducción

Hoy en día estamos viviendo la era de la Información. Las empresas necesitan estar bien informadas en el menor tiempo posible y las comunicaciones juegan un papel importantísimo.

No sólo las empresas desean estar conectadas en red, también las instituciones educativas, gubernamentales, estudiantes y profesionistas en general necesitan integrarse al mundo de las comunicaciones.

Cada usuario tiene necesidades particulares. Algunos necesitan una red de alta velocidad, otros requieren de otra red tal vez no tan rápida pero que una puntos distantes, etc.

Cada una de estas necesidades pueden ser cubiertas por distintos fabricantes, sin embargo cuando se requiere interconectar equipos de distintos fabricantes, el usuario desear que se realice en forma transparente.

Para lograr que la interconexión de equipos de distintos fabricantes sea transparente, se reunieron profesionales del área para crear estándares que mantengan uniformidad en los productos de interconexión.

La tecnología de Internet oculta los detalles de hardware y permite que las computadoras se comuniquen independientemente de las conexiones físicas de red. Esto se logra gracias a los *sistemas abiertos*. Los sistemas abiertos fueron diseñados para no depender de ningún fabricante en la implantación de un sistema de red. Cualquier fabricante que se apegue a los estándares propuestos garantiza que su producto puede convivir con cualquier otro componente del ambiente de red.

En el presente trabajo encontrará el desarrollo se presentan los temas que necesarios para la *Implantación de TCP/IP en equipos personales*, esto se aplica a los *equipos de la Facultad de Química*. La tesis en general esta orientada a la instalación de TCP/IP en los equipos personales, sin embargo, la información aquí presentada sirve como base para la implantación de TCP/IP en cualquier equipo. En el capítulo 1: "Introducción a las redes de Computadoras" se encontrarán los antecedentes, topologías y arquitecturas mas importantes para la interconexión de redes. También se da una breve descripción de las capas del Modelo OSI y del protocolo TCP/IP.

El capítulo 2, "Capa de Red (IP)", se hace un enfoque de los protocolos que intervienen en el ruteo de datos, con todos los mecanismos que esto implica, sus funciones, la fragmentación y los diversos algoritmos de ruteo que se utilizan.

El capítulo 3, "Capa de Transporte (TCP)", analiza los protocolos de transporte mas importantes, tales como TCP y UDP, también se analizan los aspectos de establecimiento de conexión y transmisión de paquetes, que hay que tomar en cuenta.

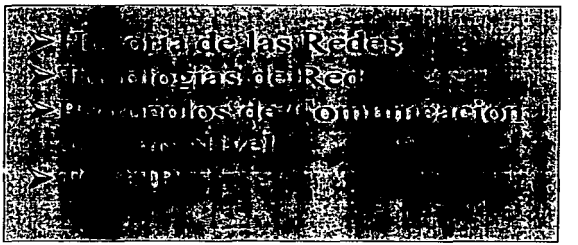
El capítulo 4, " Implantación de TCP/IP en equipo personal", describe los procedimientos para poder realizar la interconexión de equipos personales con cualquier sistema operativo.

El capítulo 5, " La red de la Facultad de Química", presenta los aspectos más importantes de la configuración de las computadoras de dicha facultad, problemas que se encontraron y las soluciones que se brindaron en cada caso. También se presenta los mapas de la infraestructura de red.

El capítulo 6: "Explotación de los recursos de Red", en este capítulo se mencionan las principales tecnologías que se apoyan en la red..

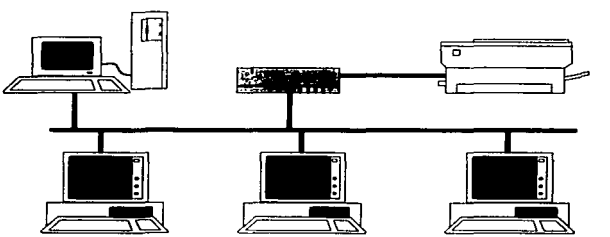
Al final del este trabajo se encontrara las fuentes de información que han sido utilizadas para la sustentación que se presenta en esta tesis.

Espero que con este breve panorama de contenido del presente trabajo, que conjunta tanto el aspecto teórico como el práctico, usted se motivo para leerlo y poder obtener una utilidad en la consulta acerca del tema del protocolo TCP/IP.



1

Introducción a las redes de Computadoras



Lista de Figuras

1.1.Historia de las redes.

Las redes surgen de la necesidad de compartir recursos de alto costo entre varias personas. En los inicios de las computadoras, los recursos eran de un costo altísimo, el poder de cómputo era uno de los mas valiosos y mas costosos. En los centros de investigación se hacia necesario que todos los investigadores tuvieran acceso a este tipo de recursos y que estos recursos fueran distribuidos en forma tal que no se tuviese un desperdicio de tiempo de procesamiento.

Las primeras redes de computadoras que se crearon, eran del tipo centralizado, es decir, un procesador central, el cual tenia el poder de cómputo y las terminales que le enviaban las tareas a realizar al procesador. Estas redes exigían que la conexión se realizara punto a punto.

El desarrollo de la tecnología permitió tener redes que comunicaban computadoras en sitios distantes, este avance obligo a crear protocolos de comunicación entre las computadoras. Estos protocolos eran propietarios de los fabricantes de las maquinas. Aquí es donde se comienza a ver la necesidad de crear de protocolos estándares para comunicar maquinas y redes de diferentes fabricantes y de diferentes tipos.

En 1973, la Agencia de Investigaciones avanzadas de la Defensa de los Estados Unidos (DARPA) inicio un programa para investigar las técnicas y las tecnologías para la interconexión de redes de diversos tipos. El objetivo era desarrollar protocolos de comunicación los cuales pueden permitir a redes de computadoras comunicarse en forma transparente a través de múltiples redes. Este proyecto fue llamado Internetting project, y el sistema de redes que emergió de estas investigaciones fue conocido como Internet.

El sistema de protocolos desarrollados en el transcurso de esta investigación, dio forma a lo que después se conocería como la suite del protocolo TCP/IP.

En 1986, la National Science Foundation (NSF) de E. U. inicio el desarrollo de la NSFNET, la cual provee un servicio de comunicación muy importante para la Internet.

La NASA y el Departamento de Energía de los E.U. contribuyeron con otra parte del canal principal de Internet con la NSINET y ESNET respectivamente. En Europa también hay una gran parte del canal principal de Internet, esta red es conocida como NORDUNET.

Actualmente Internet enlaza computadoras de Universidades, Oficinas Gubernamentales, Instituciones Públicas y Privadas, Centros de investigación, etc.

Durante el curso de la evolución de Internet, particularmente después de 1989, el sistema de Internet comenzó a integrar el soporte para otras suites de protocolos en su estructura de red básica. Se puso énfasis en una red multi-protocolo y en particular en la integración de los protocolos de Interconexión de Sistemas Abiertos (OSI, Open System Interconnection) dentro de la arquitectura.

En la década de los 80 se crearon cerca de 100 aplicaciones publicas y comerciales de protocolos de la suite TCP/IP. Durante los comienzos de 1990 también se crearon aplicaciones del protocolo OSI.

1.2. Topologías de Red.

Uno de los problemas principales de las redes de computadoras es la organización de los nodos. De esta organización depende su rendimiento y desempeño, ya que si la organización de la red es deficiente, la transmisión de datos de un lugar a otro de la red se ve disminuida, además de que en caso de que exista alguna falla, la localización del error se hace una tarea difícil.

1.2.1. Topologías básicas

Existen diferentes clasificaciones de las redes, una de ellas es la referente a la disposición física de los nodos. Con base en este criterio tenemos las siguientes topologías:

1.2.1.1. Estrella.

En una red de tipo estrella, los nodos estar organizados alrededor de un equipo central, todos los nodos están conectados a este equipo. El equipo central se encargará de atender a cada uno los nodos conectados a este. La ventaja de este tipo de configuración es que si uno de los nodos falla, todos los demás no se verán afectados, sin embargo si el equipo central falla, ninguno de los nodos podrá conectarse con los demás.

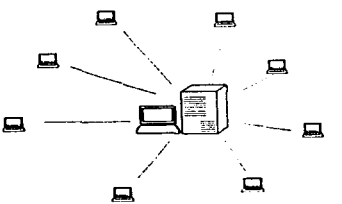


Figura 1.1. Conexión tipo estrella.
En la topología de estrella, si el servidor falla, toda la red falla.

1.2.1.2. Anillo

En esta topología la conexión de los nodos se hace uno con otro, aquí no hay un equipo central que dirija la comunicación. Simplemente se hace uso de instrumentos lógicos para controlar el acceso al medio de comunicación. La desventaja de este tipo de topologías es que si alguno de los nodos falla, el anillo completo no funciona, sin embargo en implantaciones recientes, este problema ya esta resuelto.

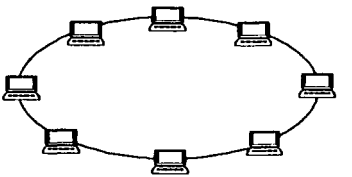


Figura 1.2. Conexión tipo anillo.
En esta topología, el punto débil es el medio de comunicación, si en algún punto, el anillo se abre, toda la red falla.

1.2.1.3. Bus

Un bus es una linea continua. Para tener acceso al bus, los nodos deben primero ver si el bus esta disponible para transmitir, es decir que ningún nodo este transmitiendo. En esta topología, si en alguno de los nodos o en alguna parte se trunca el bus, la totalidad de los nodos no podrán utilizar el bus.

En el contexto de las redes de computadoras, existen diferentes puntos en donde se debe de tener cuidado, uno como he dicho, es la disposición física de los nodos que componen a la red. Sin embargo, no solo afecta como están conectados los nodos, sino también la forma en que tendrán acceso al medio de transmisión.

Diversas organizaciones a lo largo del mundo han realizado esfuerzos para establecer estándares que regulen la organización de redes de computadoras y el acceso que tengan al medio de transmisión, entre ellas se encuentran la IEEE y la CCITT. A continuación se muestran algunos de los estándares más utilizados en el mercado de las redes.

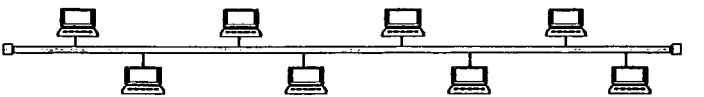


Figura 1.3. Conexión tipo bus.
Aquí, si en algún punto se rompe el bus, toda la red falla.

1.2.2. Ethernet

Ethernet es una red con estructura de BUS, con una larga historia (en términos de redes). La primera implementación de redes ethernet fue hecha en los comienzos de la década de los setenta. Esta topología adquirió gran popularidad, tuvo gran desarrollo y tiene actualmente instalados mas nodos que cualquier otra tecnología, abarcando implementaciones de mainframes, minicomputadoras y PCs. Han habido varias consideraciones para su popularidad: disponibilidad, independencia de un solo fabricante, precio bajo y simplicidad de operación e instalación.

Ethernet esta basado en la recomendación de la IEEE 802.3. El acceso a la red esta controlado por un mecanismo llamado Carrier Sense Multiple Access/Collision Detection (CSMA/CD). Este mecanismo no es único a Ethernet, pero ethernet es la mejor implementación conocida. La forma en que trabaja Ethernet es la siguiente:

- La fase de detección de portadora (Carrier Sense) es la secuencia donde un nodo checa si la red esta disponible para transmitir, el nodo "escucha" la red para ver si hay tráfico, si la red esta libre (esto es, no hay tráfico) entonces el nodo pone su mensaje en la red.
- Dado que este ejercicio lo están haciendo muchos nodos al mismo tiempo, se tiene un acceso múltiple (Multiple Access).
- Obviamente, mas de un nodo puede detectar que la red esta libre al mismo tiempo y se puede dar el caso de que más de un nodo ponga un mensaje en la red, lo cual provoca que los mensajes se encimen y pierdan significado, esto se conoce como colisión. Para monitorear esto, necesitamos la detección de colisión (Collision Detection). Se implementa en el nodo, leyendo de nueva cuenta el mensaje que acaba de ser enviado a la red, si el mensaje ha sido corrompido o fragmentado, entonces ha ocurrido una colisión.
- Cuando ocurre una colisión todo el mensaje que intentaba entrar la red es invalido y los nodos transmisores deben de tener un receso e intentar de transmitir de nueva cuenta. Para intentar transmitir de nueva cuenta, los nodos deben de esperar un tiempo aleatorio, esto es para evitar que se repita la colisión.

La topología de Ethernet tradicionalmente ha sido cableada con cable coaxial; el cable Ethernet grueso (También llamado Thick Ethernet de mas o menos 20 mm de diámetro) permite tener un segmento de hasta 500 metros (con un máximo de 250 nodos a intervalos de 2 metros). Cuando se requiere de una longitud mayor, se pueden usar repetidores para las extensiones, la cantidad máxima de segmentos permitidos es de 5, es decir, con Ethernet se pueden alcanzar longitudes de hasta 2.5 Km

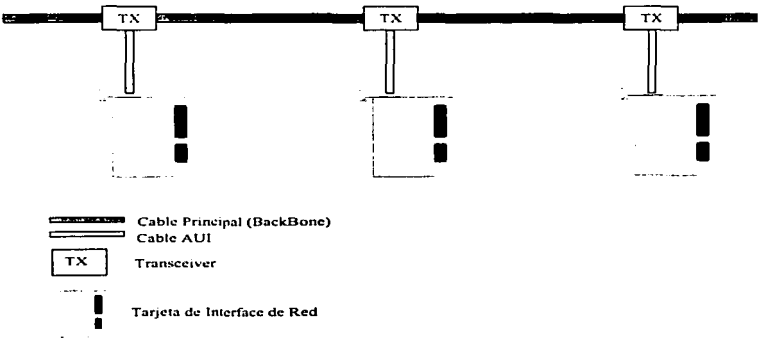


Figura 1.4. Conexión de Thick Ethernet.
En este tipo de conexión, se utiliza un transceiver, al cual se le conoce como el nombre de vampiro, ya que virtualmente muerde al cable coaxial para hacer contacto con los contactos alambricos.

El cable coaxial delgado (denominado Thin Ethernet y comúnmente "Cheapernet") usa un cable coaxial semejante al de televisión (mas o menos 6 mm de diámetro) y permite tener una longitud máxima de 185 metros, al igual que con el cable coaxial grueso, se pueden utilizar extensiones hasta alcanzar un total de 3 segmentos, aproximadamente 555 metros. Cada uno de los segmentos de cable coaxial delgado puede tener un máximo de 30 nodos conectados.

Para la conexión del cable con la tarjeta de red, en Thin Ethernet se utilizan conectores tipo "T" y en Thck Ethernet se utilizan convertidores de cable coaxial grueso a conector tipo AUI (Attachment Unit Interface) que es el conector universal para tarjetas de red.

Uno de los problemas más grandes que tiene ethernet con cable coaxial es que es común que las uniones entre el cable y el conector fallen. Dado que es una topología de bus, si uno de los nodos falla, toda la red falla y para encontrar el error en una red de grandes dimensiones, se torna en un trabajo difícil.

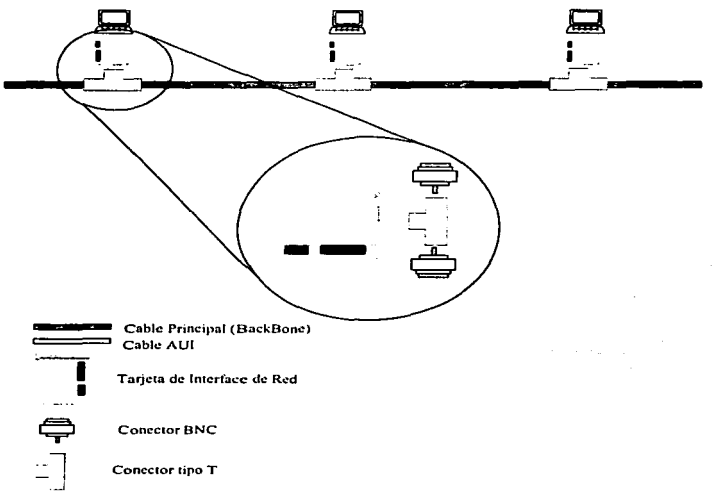


Figura 1.5. Conexión de Thin Ethernet.
En este tipo de conexión, se requiere del uso de conectores tipo T, los cuales permiten que la señal pase a través del cable coaxial y pase también a la tarjeta de interfaz de red.

El desarrollo de la tecnología ha dado como fruto concentradores ethernet (hubs). Los concentradores utilizan cable UTP (Unshielded Twisted Pair) para conectar los nodos al concentrador. Las características de este tipo de cableado ha sido definido por el estándar 10-Base-T de la IEEE y mantiene la velocidad de 10 Mbps.

Con instalaciones de UTP cada nodo puede alcanzar un máximo de 100 metros desde el concentrador hasta la maquina. Físicamente el bus se convierte en una estrella con esta tecnología.

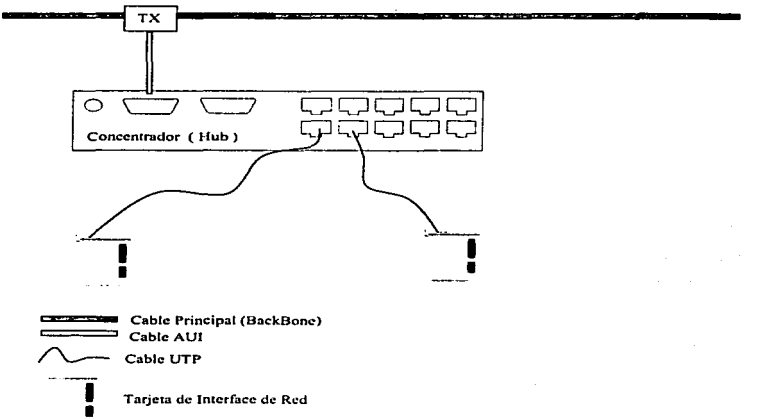


Figura 1.6. Cableado Ethernet 10-Base-T.
En este tipo de cableado, la topología lógica es de tipo bus, pero la topología física es de forma de estrella.

1.2.3. Token Ring.

El concepto de un sistema de control Token Ring para Redes de área local tiene mas de 20 años de existencia, pero el producto no figuro en el mercado de las redes de área local con PC's hasta que IBM comenzó a promover las redes Token Ring en 1986. Para esas fechas, el mercado de las redes de área local ya estaba establecido, y Ethernet junto con varios sistemas propietarios tenían una gran parte del mismo.

Netware dominaba gran parte del mercado de las redes y los productos de IBM tenían que competir con él, así que cuando IBM introdujo Hardware Token Ring, también promociono su propio software (el programa PC-LAN) basado en MS-DOS. La falta de aceptación del softtware tuvo un impacto también en la

aceptación del hardware Token Ring. Esto, además del costo adicional y la complejidad de diseño e instalación de redes de área local Token Ring, y la aparente velocidad inferior de Token Ring comparado con Ethernet (4 Mbps contra 10 Mbps), significaron que Ethernet continuara siendo la topología dominante en el mercado de las redes de área local de PC's.

Las especificaciones de redes Token Ring están definidas en los estándares 802.2 y 802.5 de la IEEE. Versiones posteriores han derivado en velocidades de transmisión de 16 Mbps, sin embargo, la implementación mas utilizada es la de 4 Mbps debido a que los costos se incrementan además de que pueden existir problemas con redes de gran tamaño.

Token Ring opera en 4 Mbps o 16 Mbps; estas dos velocidades de transmisión no pueden ser mezcladas dentro de un mismo anillo. Una red Token Ring solo puede tener un mensaje unido a la estafeta (token) en la red a un tiempo.

Las redes Token Ring son consideradas lógicamente como un anillo, pero son físicamente construidas como una serie de redes estrella ligadas. Ver Figura 1.7. Cada nodo en la red esta conectado al anillo con uniones de entrada y de salida cada uno. Si un nodo falla, entonces la unión a un nodo puede ser cerrada para mantener la integridad del anillo para otros usuarios.

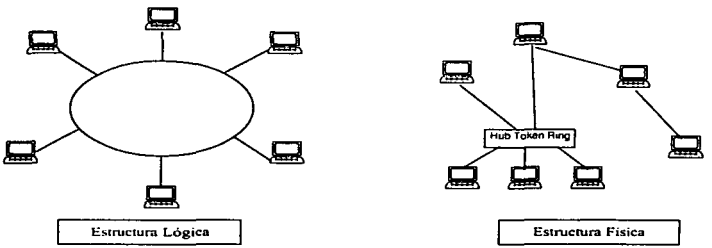


Figura 1.7. Estructura física y lógica de una red tipo anillo.
Físicamente el anillo puede estar constituido por un conjunto de redes tipo estrella, sin embargo, lógicamente será tratado como un anillo.

Existen además de estos dos estándares, muchos otros para la disposición de nodos y acceso al medio de transmisión. La IEEE es una de las organizaciones más importantes en el desarrollo de estándares de comunicación de redes de computadoras.

Algunos de los sub-comites de la IEEE mas importantes para el desarrollo de estándares son:

- Sub-comite 802.1 Define la interfaz de mas alto nivel.
- Sub-comite 802.2 Define el control lógico de enlace.
- Sub-comite 802.3 Define redes CSMA/CD, tales como Ethernet.
- Sub-comite 802.4 Define redes Token Bus.
- Sub-comite 802.5 Define redes Token Ring
- Sub-comite 802.6 Define redes de área metropolitana (MAN)
- Sub-comite 802.7 Define redes de banda ancha.
- Sub-comite 802.8 Define estándares de fibra óptica
- Sub-comite 802.0 Define redes integradas de voz y datos.

Ya he hablado de los estándares de organización y acceso al medio mas utilizados en el mundo de las redes, sin embargo ahora surge la dificultad de conectar computadoras de diferentes plataformas y de distintos fabricantes, además de estar conectadas con diferentes estándares.

Este problema se ha podido solucionar gracias a los protocolos de alto nivel.

1.3. Protocolos de Comunicación de Alto Nivel.

La Organización Internacional para Estandarización (ISO) desarrolló un modelo de referencia para la estandarización de los protocolos de red. El modelo es conocido como el modelo de referencia para Interconexión de Sistemas Abiertos, (OSI, Open System Interconection). OSI es un modelo de siete capas.

La forma en la cual dos partes de la red se comunican es llamada protocolo, lo cual asegura que cada una de las partes de la comunicación entienda a la otra sin ambigüedad. Un protocolo puede especificar la forma en que los datos son codificados, como puede ser identificado el comienzo y el fin de un mensaje, como las direcciones de los puntos origen y destino son mostradas, y las acciones a tomar si se encuentran errores durante la transmisión. Debe existir un protocolo definido para conectar dos niveles adyacentes, pero la estructura completa de una red puede consistir de muchas especificaciones diferentes.

Por ejemplo, la capa de red (Capa 3), tendrá especificaciones de protocolo para la capa de enlace de datos (capa 2) y para la capa de transporte (Capa 4). Esto es, a la capa 3 no le interesan las especificaciones para la capa 1 o para la capa 5, dado que esas capas no forman parte del área de interés de la capa. Esto da como consecuencia una flexibilidad considerable cuando las especificaciones son cambiadas o se agregan nuevas opciones.

Figura 1.8. Modelo OSI.
El modelo OSI sólo se ha podido implantar en algunos sistemas experimentales debido al alto costo que representa su implementación.

1.3.1. Detalles del modelo de las siete capas de OSI.

1.3.1.1. Capa Física.

Esta capa describe las especificaciones mecánicas y eléctricas para la estructura del cableado. define como serán convertidos los bits en corriente eléctrica, pulsos luminosos o cualquier otra forma física. Determina el método por el cual las ráfagas de bits son enviadas a través de la red. Esta capa proporciona los servicios de enlace que están asociados con la adquisición, mantenimiento y desconexión de circuitos físicos que conforman la ruta de conexión de la comunicación. Maneja tanto la interfaz como los requerimientos procedurales del medio de conexión. La capa física es similar a la interfaz DCE-DTE. La capa física esta encargada de la sincronización de los bits y de la identificación de un elemento como un uno o un cero. La unidad de datos de este protocolo es el Bit.

Protocolos típicos en la capa física incluyen la familia RS-232, la familia RS-449, las interfaces CCITT X.25y X.21, otra serie de recomendaciones de la CCITT (V y X), y los aspectos físicos de los protocolos de acceso al medio de la IEEE 802.X para redes de área local.

En los estándares de cableado de cobre en redes Ethernet existen dos tendencias básicas: cableado coaxial y cable UTP (Unshielded Twisted Pair). El cableado coaxial tiene básicamente dos variantes: el cable coaxial delgado y el cable coaxial grueso.

1.3.1.2. Capa de Enlace de datos.

Esta es la responsable de hacer que el enlace físico sea confiable. Esta encargada de comenzar y terminar los enlaces, además de detectar y controlar los errores. El trabajo del comité IEEE 802.3 ha subdividido esta capa en dos subcapas.

La capa que sirve como interfaz con la capa física es llamada capa de Control de Acceso al Medio (MAC) y la capa que sirve de interfaz con la capa de transporte es llamada capa de Control Lógico de Enlace.

La capa de control lógico de enlace es responsable de ensamblar y particionar los frames, agregando direcciones origen y destino, facilidades para el control y detección de errores en el receptor. Los Puentes (Bridges) funcionan en esta capa.

LLC es la capa responsable de controlar el intercambio de datos entre usuarios que se están comunicando a través de la capa de Control de Acceso al Medio.

Con excepción de la capa física, los servicios y protocolos proporcionados por la capa de enlace deben de ser familiares a aquellos que están en la industria de la comunicación de datos. Los servicios de la capa de enlace están relacionados con el intercambio confiable de datos a través de un enlace punto a punto o multipunto que ha sido establecido en la capa física. Los protocolos de la capa de enlace de datos manejan el establecimiento, control y terminación de la conexión lógica.

Controla el flujo de datos del usuario, supervisa la recuperación de errores y condiciones anormales, mantiene la sincronización de los bloques o frames y caracteres.

1.3.1.3. Capa de Red.

La capa de red provee aquellos servicios asociados con el traslado de los datos de los usuarios a través de una red constituida por enlaces encadenados, teniendo muchas rutas disponibles entre los puntos. Estos servicios incluyen ruteo, switcheo, secuenciación de datos, control de flujo y recuperación de errores. Funciones como control de flujo y recuperación de errores aparecen duplicados en el nivel de enlace, estos están relacionados con conexiones a través de múltiples enlaces.

Esta capa es la responsable de establecer y monitorear las conexiones entre redes de área local. Esta capa es independiente de la capa física, es decir, esta capa puede estar sobre cualquier protocolo de capa 2.

En esta capa se realiza el control y selección de las rutas lógicas y conexiones entre usuarios de puntos finales en una red. Un ejemplo seria un circuito virtual en una red publica de datos.

Los capa de paquetes del CCITT X.25 es el mejor protocolo de capa de red para redes de switcheo de paquetes. X.21 es usado para redes de switcheo de circuitos. El Departamento de Defensa de E.U. ha desarrollado un protocolo de internet conocido como IP. Otros ejemplos de protocolos de red incluyen el CCITT Q.931 y el protocolo ISO 8473 no orientado a conexión, después veremos la diferencia entre un servicio orientado a conexión y uno no orientado a conexión.

1.3.1.4. Capa de Transporte.

La capa de transporte es la capa mas alta asociada con el movimiento de datos a través de la red. Esta capa provee un mecanismo universal transparente para ser usado por las capas mas altas que representa a los usuarios de los servicios de comunicación. De la capa de Transporte se espera la optimización del uso de los recursos disponibles.

Los protocolos de transporte son responsables de la integridad del intercambio de datos y deben de ser el puente conector entre los servicios proporcionados por las capas inferiores y los requeridos por las capas superiores. Se han desarrollado numerosas clases de protocolos de transporte desde algunas muy simples hasta otras muy complejas. Las capas de transporte simples pueden ser utilizadas cuando la red provee un servicio confiable y de calidad.

Clase	Nombre	Tipo de Red	Características
0	Clase Simple	Tasa aceptable de errores residuales (no detectables). Línea Dedicada confiable o red de conmutación de paquetes confiable.	No multiplexa, No se Recupera de Errores reportados por la capa de red, no detecta ni se recupera de errores no reportados por la capa de red.
1	Clase de recuperación básica de errores	Tasa aceptable de errores detectados. Tasa aceptable de errores residuales(No detectados), tasa no aceptable de errores detectados. Red de conmutación de paquetes no confiable.	No multiplexa, se recupera de errores reportados por la capa de red, no detecta ni se recupera de errores no reportados por la capa de red.
2	Clase de Multiplexaje	Tasa aceptable de errores residuales (no detectables) Línea Dedicada confiable o red de conmutación de paquetes confiable.	Multiplexa, no se recupera de errores reportados por la capa de red, no detecta ni se recupera de errores no reportados por la capa de red.
3	Clase de Multiplexaje y recuperación de errores	Tasa aceptable de errores detectados . Tasa aceptable de errores residuales(No detectados), tasa no aceptable de errores detectados. Red de conmutación de paquetes no confiable.	Multiplexa, se recupera de errores reportados por la capa de red, no detecta ni se recupera de errores no reportados por la capa de red.
4	Clase de detección y recuperación de errores.	Tasas no aceptables de errores detectados y no detectados. Red no orientada a conección.	Multiplexa, divide la conexión de transporte entre muchas conexiones de red, permite el uso de redes no orientadas a conexión, recuperación de errores reportados por la capa de red, detección y corrección de errores no reportados por la capa de red.

Figura 1.9. Clases de protocolos de transporte.

Un protocolo de transporte complejo es usado cuando los servicios de las capa inferior es incapaz de proporcionar el nivel de servicio requerido. La complejidad es necesaria debido a que esta capa duplica los mecanismos de recuperación que deben haber sido proporcionados por las capas inferiores.

La ISO ha promulgado el estándar internacional 8073 como un protocolo de transporte. Este estándar define 5 clases de protocolos, desde el mas simple (clase 0) hasta el mas complejo (clase 4).

Un ejemplo de protocolo de transporte es TCP, desarrollado por el Departamento de Defensa de los E.U y que forma parte de la suite de protocolos de TCP/IP

1.3.1.5. Capa de Sesión.

Una sesión enlaza dos procesos de aplicación en una relación cooperativa durante cierto tiempo. La capa de sesión proporciona un servicio administrativo que maneja el establecimiento y liberación de una conexión entre dos entidades de presentación. Las sesiones son establecidas cuando un proceso de aplicación pide acceso a otro proceso de aplicación.

Cuando una sesión es establecida, los servicios de control dialogan y supervisan el intercambio de datos actual. El propósito de esta capa es proporcionar el control sobre la comunicación entre las aplicaciones. Esta asume que la conexión física es confiable y es controlada por las capas inferiores. Una simple sesión puede mantener varias conexiones de transporte o muchas sesiones consecutivas pueden ser mantenidas en una conexión de transporte única. Actualmente los protocolos de sesión incluyen el ISO 8327, el CCITT X.25, ECMA 75 y el CCITT T.62 el cual esta orientado a servicios de Teletex.

1.3.1.6. Capa de Presentación.

Esta capa permite a una aplicación interpretar en forma adecuada la información transferida. Esta capa esta involucrada con la traducción, transformación, formato y sintaxis de la información. Esas funciones son requeridas para adaptar las características de manejo de la información de un proceso de aplicación a otro. Algunos ejemplos de las acciones que se realizan en esta capa, serian, la traducción de códigos, estructuración de los datos para el despliegue en pantalla, control de formato y protocolos de terminales virtuales.

Esta capa es la responsable de presentar los datos a aplicaciones diferentes en un formato que ambos puedan reconocer. También controla características tales como cifrado y compresión de datos. Un ejemplo de la función de esta capa es la de convertir datos ASCII, usados por la mayoría de las PCs y el sistema de códigos EBCDIC usado en las mainframes IBM.

La ISO realizo una selección internacional de estándares de presentación, conocido como DIS 8823. La representación sintáctica de datos ha sido definida en DIS 8824 y 8825. La CCITT ha descrito el protocolo de presentación para manejo de mensajes en X.409 y para Telex en X.61.

1.3.1.7. Capa de Aplicación.

Incluye una parte de la administración de la red y tareas de aplicación general, tales como transferencia de archivos. Aunque esta es la capa superior de la arquitectura del modelo OSI, la capa de aplicación no es la casa de las aplicaciones. Esta es simplemente la ventana a través de la cual las aplicaciones obtienen el acceso a los servicios proporcionados por la arquitectura de comunicaciones.

Esta capa proporciona servicios de comunicación que son mas directamente comprensibles al usuario. Estas incluyen identificación de procesos cooperativos, autenticación del comunicante, verificación de autoridad, determinación de los recursos disponibles y acuerdo de sintaxis.

La capa de aplicación puede ser visualizada como una conexión de elementos de usuario que son específicos al proceso de aplicación; un elemento de aplicación especifica tiene funciones como transferencia de archivos, intercambio de datos de negocio, o operaciones de terminales virtuales y un elemento común constituido de funciones generales.

Aunque el Modelo OSI no ha sido implantado en forma comercial y su uso no sea mas que teórico, lo tomaremos como referencia para ubicar las funciones de TCP/IP.

1.4.TCP/IP

La suite de protocolos permite a computadoras de todos los tamaños, de diferentes proveedores, corriendo sistemas operativos totalmente diferentes comunicarse entre si. TCP/IP comenzó a finales de los 60's como un proyecto de investigación financiado por el gobierno de E. U.

TCP/IP es realmente un sistema abierto en el cual la definición de la suite de protocolos y muchas de sus implementaciones están disponibles al publico con un pequeño o nulo cargo. También le da forma a lo que se le llama World Wide Internet o la Internet.

Como ya hemos visto, los protocolos de red son desarrollados normalmente en capas, cada capa es responsable de una faceta diferente de la comunicación. Una suite de protocolos como TCP/IP es la combinación de diferentes protocolos en varias capas. TCP/IP es normalmente considerado como un sistema de cuatro capas, como se muestra en la Figura 1.10.

Aplicación	Telnet, ftp, e-mail, etc.
Transporte	TCP, UDP
Red	IP, ICMP, IGMP
Enlace	Controlador del dispositivo y tarjeta interfaz.

Figura 1.10. Las cuatro capas de la suite de protocolos de TCP/IP.
La suite de protocolos de TCP/IP ha probado ser muy eficaz al interconectar computadoras heterogéneas entre si, esto se debe a su independencia entre capas.

1.4.1. Capa de Enlace

Algunas veces es llamada también capa de enlace de datos o la capa de interfaz de red. Normalmente incluye el controlador del dispositivo en el sistema operativo y la correspondiente interfaz de red en la computadora. Juntos, manejan todos los detalles de la interfaz física con el cable (o cualquiera que sea el tipo de media que este siendo usado)

1.4.2. Capa de Red

También conocida como capa de internet, maneja el movimiento de paquetes a través de la red. Por ejemplo, el ruteo de paquetes es realizado aquí. IP (Internet Protocol), ICMP (Internet Control Message Protocol) y IGMP (Internet Group Management Protocol) proporcionan la capa de red en la suite TCP/IP.

1.4.3. Capa de transporte

La capa de transporte proporciona un flujo de datos entre dos computadoras. En la suite del protocolo TCP/IP hay dos protocolos de transporte totalmente diferentes: TCP (Transmission Control Protocol) y UDP (User Datagram Protocol).

TCP proporciona un flujo confiable de datos entre dos hosts. Esta relacionado con cosas tales como la división de datos de una aplicación en trozos de tamaño apropiado a la capa de red adyacente, y proporciona mecanismos de control para la recepción y envío de paquetes, configura los tiempos de fin de vida de un paquete, etc. Dado que este flujo de datos confiable, la capa de aplicación puede ignorar todos estos detalles.

UDP por otro lado, proporciona un servicio mucho mas simple a la capa de aplicación. Este solo manda paquetes de datos llamados datagramas desde un host a otro, pero no hay garantía de que el datagrama llegue al otro host. Cualquier implantación de confiabilidad debe ser agregada por la capa de aplicación.

Existe un uso para cada tipo de protocolo de transporte, el cual veremos cuando veamos las diferentes aplicaciones que usan TCP y UDP.

1.4.4. Capa de Aplicación.

Esta capa es la encargada de manejar los detalles de la aplicación en particular. Hay muchas aplicaciones comunes de TCP/IP que al menos cada implementación proporciona:

- Telnet para sesiones remotas
- FTP. Protocolo de Transferencia de Archivos. (File Transfer Protocol).
- SMTP. Protocolo Simple de Transferencia de Correo (Simple Mail Tranfer Protocol) para correo electrónico.
- SNMP. Protocolo de Administración de Redes Simples. (Simple Network Management Protocol).

Por ejemplo, si tenemos dos hosts en una red de área local (LAN), tal como ethernet, ambos corriendo FTP, la interacción de protocolos se ve como en la Figura 1.11. Aquí se nota que la capa de aplicación solo se encarga de los detalles propios de la aplicación, no se encarga de ver los detalles de la transmisión.

Hay que notar que en la capa de aplicación, un servicio de ftp es cliente y el otro es servidor. La mayoría de las aplicaciones de red están diseñadas de forma que en uno de los extremos hay un cliente y en el otro el servidor. El servidor proporciona algún tipo de servicio a los clientes, en este caso acceso a los archivos en el servidor.

Cada capa tiene uno o mas protocolos para comunicarse con su contraparte en la misma capa. Un protocolo, por ejemplo, permite a dos capas de TCP comunicarse y otro protocolo permite a dos capas de IP comunicarse.

Hay una diferencia crítica entre la capa superior de la Figura 1.11 y las tres capas inferiores. La capa de aplicación esta interesada solo por los detalles de la aplicación y no con el movimiento de datos a través de la red. Las capas inferiores no saben nada de la aplicación pero manejan todos los detalles de la comunicación.

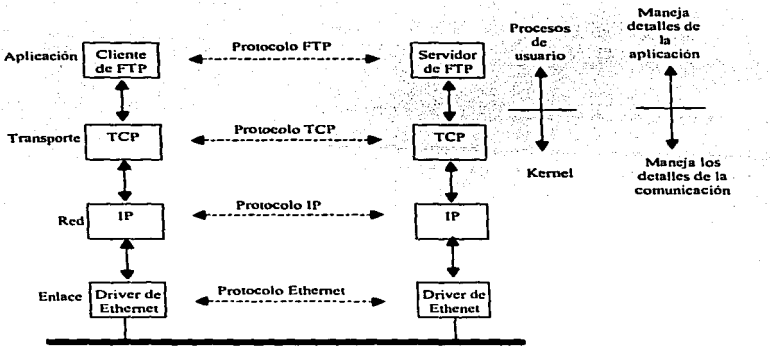


Figura 1.11. Dos hosts en una Red de Area Local
Aunque la capa de transporte se conecta con la capa de red, lógicamente, cada capa se comunica con su correspondiente en el otro extremo de la comunicación

Aunque en la Figura 1.11, solo aparece un protocolo por capa, la realidad es que pueden existir varios protocolos para la misma capa. La principal razón que dio origen a la creación de este tipo de protocolos , fue la de interconectar muchas redes entre si, sin que se perdiera el control sobre la comunicación.

Estas redes podrían tener diferentes protocolos de bajo nivel, una con ethernet, otra con token ring, etc. Para conectar estas redes y para distinguir el tráfico interno del que hay entre redes, existen dispositivos llamados ruteadores. Un ruteador por definición tiene dos o mas capas de interfaz de red.

Un esquema de funcionamiento de un ruteador se muestra en la figura1.12.

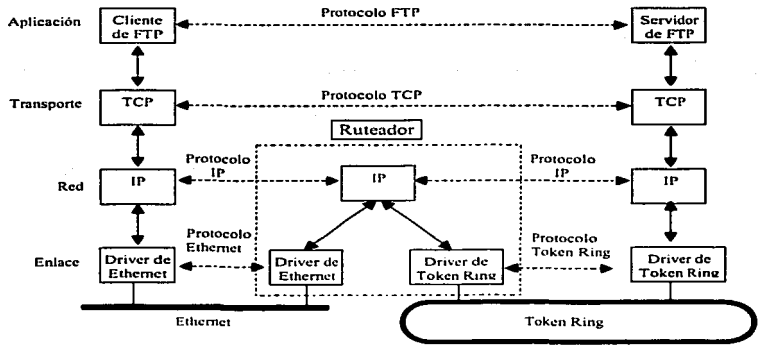


Figura 1.12. Dos redes conectadas con un ruteador.
El ruteador debe distinguir entre el trafico de cada red, separa los datagramas enviados hacia entidades de una misma red y los enviados hacia entidades de la otra red

En la suite de protocolo TCP/IP, la capa de red, IP, proporciona un servicio poco confiable. Esto es, Hace su mejor trabajo moviendo datos desde el origen hasta el destino, pero no hay garantías. Sin embargo, TCP, proporciona una capa de transporte confiable usando el servicio no confiable de IP. Para proporcionar este servicio, TCP establece tiempos de espera y retransmisiones, envía y recibe avisos de recepción de datagramas, etc.

UDP por otro lado, proporciona servicios en la capa de transporte que no son orientados a conexión (connectionless). A través de UDP, la comunicación ya no solo se define en base a direcciones de computadoras, sino de aplicaciones dentro de esas computadoras.

UDP no garantiza ningún nivel de servicio, no ordena los mensajes, no cuenta con mecanismos de control de flujo, mecanismos contra la duplicación ni contra la perdida de mensajes. Simplemente sirve para que las aplicaciones tengan puntos de entrada a un servicio de transferencia de datos a través de la red, ya que las capas inferiores no proporcionan el esquema de direccionamiento necesario para llevar esto a cabo.

Como ya hemos dicho, existen muchos protocolos en la suite de TCP/IP. La siguiente figura muestra algunos de los protocolos adicionales que son ocupados:

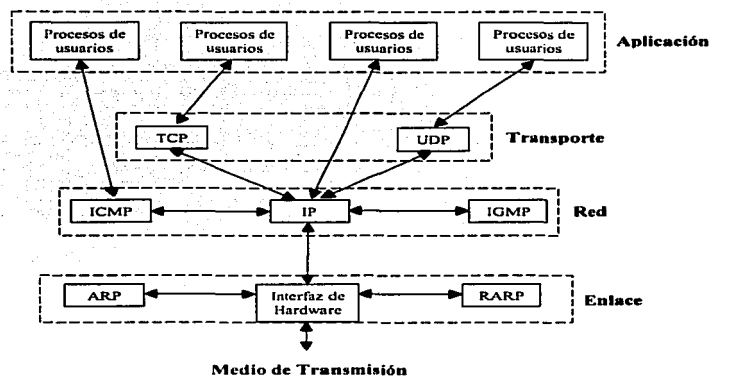


Figura 1.13. Varios protocolos en las diferentes capas en la suite de protocolos de TCP/IP. *No existe un solo protocolo para cada una de las capas, como podemos ver, existen varios protocolos que desempeñan tareas complementarias.*

Ya hemos visto como se comunican las capas de la suite de protocolos de TCP/IP, pero ahora veamos como esta relacionado el modelo OSI con esta suite. Aunque OSI es un estándar que no ha sido llevado a la práctica, ha servido como guía para ir llevando a las suites de protocolos hacia una estandarización paulatina, en la figura 1.14, además de mostrar la relación de OSI con TCP/IP, también mostraré su relación con otros estándares de comunicación.

La red Internet se caracteriza por comunicar sistemas tan distintos como los que se ven en la figura anterior. Para que se puedan comunicar en Internet, cada interfaz debe tener una dirección Internet única (también llamada dirección IP). Estas direcciones son numero de 32 bits. En lugar de tener un espacio de direcciones plano como 1,2,3, etc., hay una estructura de direcciones IP.

Capas OSI	Sistemas Apple	Sistemas Banyan	DEC DECNET	IBM SNA	Redes Microsoft	Novel Netware	TCP/IP	Xerox XNS
Aplicacion	Programas de aplicacion y protocolos para la transferencia de archivos							
Presentacion	Apple Talk Filing Protocol	Remote Procedural Calls	Administracion de redes y aplicaciones de red	Servicios de transacción Presentacion de servicios	Server Message Block (SMB)	Protocolos nucleos de Netware	Protocolos de Aplicacion especifica.	Interaccion de procesos y control.
Sesion	Protocolo de sesion Apple Talk	Remote Procedure Calls	Sesion	Control de Flujo de Datos	NetBios	NetBios	Telnet, FTP, SNMTP, etc.	Interacción de procesos y control
Transporte	Protocolo de Transacciones Apple Talk (ATP)	Comunicacion interprocesos VINES	Comunicaciones finales	Control de Transmision	NetBEUI	SPX	TCP	SSP
RED	Protocolo de envio de datagramas (DDP)	Protocolo de Internet VINES	Ruteo	Control de Ruta	NetBEUI	IPX	IP	IDP
Enlace de Datos	Tarjetas de Interfaz de Red: Ethernet, Token-Ring, etc Controladores de Tarjetas ODI, NDIS, etc							
Fisica	Medio de transmision Par tensado, Coaxial, Fibra optica, Microondas, etc							

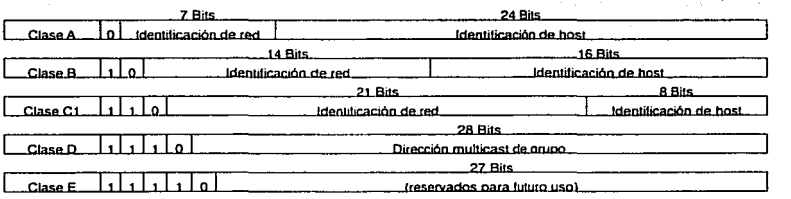
Figura 1.14. Relación del modelo OSI con otros estándares.

La Figura 1.15 Muestra las cinco diferentes clases de direcciones Internet y el rango que abarcan.

Existe una autoridad que asigna los rangos de direcciones a nivel mundial, esta autoridad es la Internet Network information Center, conocida como InterNIC.

La mayor parte de los protocolos que se establecen primero deben pasar por la prueba del mercado. Idealmente, este proceso es a la inversa, es decir, primero se genera el estándar, después se generan los prototipos, se prueban y después se lanzan al mercado.

La mayor parte de los estándares definidos se basan en implantaciones existentes. Por ejemplo, TCP/IP nació de los productos que fueron lanzados al mercado por instituciones gubernamentales o institucionales. Pero aquí surge la duda, ¿Quién aprueba los nuevos estándares? ¿Quién propone las bases para los nuevos estándares?



Clase	Rango
A	0.0.0.0 A 127.255.255.255
B	128.0.0.0 A 191.255.255.255
C	192.0.0.0 A 223.255.255.255
D	224.0.0.0 A 239.255.255.255
E	240.0.0.0 A 247.255.255.255

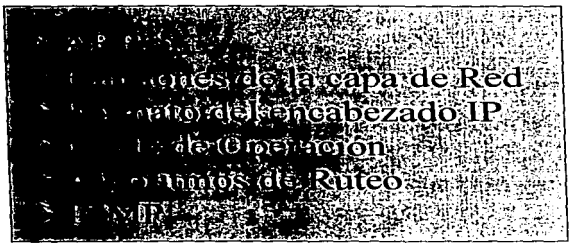
Figura 1.15. Formato de direcciones IP.

Existen cuatro grupos responsables de la tecnología de Internet:

- La Internet Society (ISOC) es una asociación profesional para facilitar, dar soporte y promover la evolución y crecimiento de Internet como una infraestructura de investigación global de comunicaciones.
- La Internet Architecture Board (IAB) es el cuerpo de coordinación y revisión técnica. Esta compuesta de aproximadamente 15 voluntarios internacionales de varias disciplinas y funciona como la mesa final editorial y de revisión técnica para la calidad de los estándares de Internet. La IAB esta bajo la ISOC.
- La Internet Engineering Task Force (IETF) esta formada de areas específicas (aplicaciones, ruteo y direccionamiento, seguridad, etc), desarrolla las especificaciones que se convierten en estandares de Internet.
- La Internet Research Task Force (IRTF) desarrolla proyectos de investigación de largo alcance.

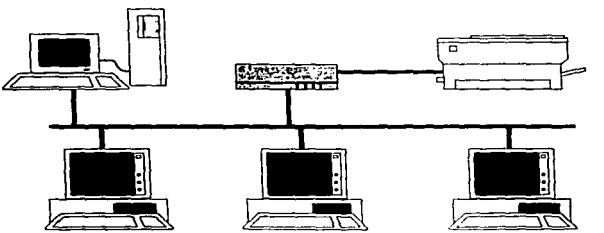
Tanto la IRTF como la IETF están debajo de la IAB.

Todos los estándares en la comunidad de Internet son publicados como un RFC (Request For Comment). Además en los RFC hay estándares no oficiales, pero son publicados para propósitos de información. El rango de tamaños de los RFC's varían de 1 hasta 200 páginas. Cada RFC es identificado con un número, tales como RFC1122.



2

La capa de Red (IP)



Lista de Figuras

2.1. ARP

Hasta este momento he descrito el esquema de direcciones de TCP/IP, en este esquema, cada host tiene asignada una dirección de 32 bits, esto le permite a la red Internet comportarse como una red virtual.

También he mostrado algunas topologías físicas de redes. Se debe notar que un host en una red física dada, se pude comunicar con otro solo si conoce la dirección física de la otra máquina, sin embargo, no hay una relación directa entre la dirección IP de 32 bits y la dirección física de red, además de que la dirección física esta intrínsecamente ligada a la tecnología de red. Existen diversos mecanismos para realizar la conversión entre una dirección física y una dirección IP. Estos mecanismos tienen por objetivo ocultar la complejidad del mapeo entre direcciones para permitir a las aplicaciones de mas alto nivel trabajar solamente con direcciones IP. Sin embargo, se debe de tener en cuenta que en realidad toda la comunicación al nivel de hardware se tiene que realizar con direcciones físicas.

ARP (Address Resolution Protocol) es un protocolo que permite conocer la dirección física de una máquina correspondiente a una dirección IP dentro de una misma red física.

El mecanismo con el que trabaja este protocolo es muy simple. Si alguna estación quiere saber la dirección física de otra estación, envía un mensaje de tipo broadcast a todos los hosts sobre una misma red física, con la dirección IP de la máquina deseada, este mensaje de formato especial, realiza la pregunta ¿Quien tiene asignada esta dirección IP?. Entonces el dueño de la dirección IP envía un mensaje con su dirección física.

Este mecanismo resulta de alto costo para la red, cada máquina mantiene un cache de direcciones IP con sus correspondientes direcciones físicas. Así, cada vez que se desee enviar un paquete por la red, la máquina deberá consultar primero el cache y verificar si ya existe la dirección física de la máquina asociada a una dirección IP.

ARP reside en la capa de interfaz de red. ARP enlaza direcciones de alto nivel, como IP, y direcciones de bajo nivel, direcciones físicas. El software de enlace forma un limite entre las capas más altas del software de protocolo los cuales solo usan direcciones IP y las capas inferiores de controladores de dispositivos, los cuales solo utilizan direcciones físicas. ARP encapsula el mapeo de tablas y maneja tanto el chequeo de las tablas como la actualización de las mismas.

Conceptualmente ARP puede ser dividido en tres partes, una sección de salida, una de entrada y un administrador de cache. Cuando un datagrama es enviado, el software de interfaz de red llama la sección de salida para enlazar una dirección de algún protocolo de alto nivel (por ejemplo IP) a su correspondiente dirección de hardware.

Los procedimientos de la sección de salida regresan el enlace, el cual es usado por las rutinas de interfaz de red para encapsular y transmitir el paquete. La sección de entrada maneja los paquetes de ARP que llegan de la red. Los procedimientos de la sección de entrada actualizan el cache de ARP agregando nuevos enlaces. El administrador del cache implanta las políticas de remplazo. Examina las entradas en el cache y las retira cuando han cumplido un tiempo determinado.

Al construir software de ARP se debe de tener cuidado en implantar todos los aspectos del protocolo, ya que algunas aplicaciones pueden dar enlaces incorrectos debido a que eliminan el temporizador del cache con el fin de dar mayor eficiencia. Algunos parámetros que se deben de tener en cuenta cuando se desarrolla un software de ARP son los siguientes:

- Tamaño del cache de ARP
- Intervalo de tiempo que el cliente esperará una respuesta de ARP
- Número de veces que un cliente volverá a intentar una petición.
- Intervalo de tiempo entre intentos.
- Tiempo de vida de una entrada en el cache
- Tamaño de la cola de retransmisión de paquetes.

Los diseños típicos usan constantes simbólicas para parámetros tales como el cache, permitiendo al administrador del sistema cambiar la configuración para instalaciones específicas. Para instalaciones en las cuales los administradores necesitan mas control, se pueden escribir programas de utilerías que permitan al administrador realizar cambios aun cuando la aplicación este corriendo. Por ejemplo, software que sea capaz de examinar el cache de ARP, borrar una entrada o cambiar valores (por ejemplo el campo de tiempo de vida), etc., sin embargo, algunos campos no pueden ser cambiados fácilmente. Por ejemplo, muchos programadores eligen tiempos de retransmisión mixtos o exponenciales y esto lo implantan directamente dentro del código.

La abstracción de la interfaz de red define la interfaz entre el software de protocolo en el sistema operativo y el hardware que lo soporta. Esta oculta los detalles de hardware y permite al software de protocolo interactuar con una amplia variedad de hardware de red utilizando las mismas estructuras de datos.

2.2. *Funciones de la capa de red.*

La capa de red se encarga del ruteo, conmutación e intercambio de información entre redes, con el fin de proporcionar un camino entre los puntos finales (de red) en una transmisión de datos que sea uniforme, independientemente de las redes físicas que se utilicen para llevar esto a cabo.

La capa lógica de red opera sobre múltiples redes físicas, estas redes pueden ser de tecnología diferente (redes de área local, Redes X.25, ISDN, etc.). En la terminología OSI, estas redes son llamadas subredes (subnetworks). Las redes X.25 son en términos generales redes para acceso telefónico.

Las redes ISDN es una red formada por redes de varios tipos, redes de datos, de voz, de vídeo, etc.

IP es el protocolo por el cual las suite de protocolos de TCP/IP se mueve, tanto TCP, UDP, ICMP e IGMP se transmiten como datagramas de IP. (Ver figura 1.13) Un hecho que sorprende a muchos recién iniciados en TCP/IP especialmente aquellos que tienen antecedentes de uso en redes X.25 y SNA es que IP proporciona un servicio de envío de datagramas no orientados a conexiones poco confiable.

Por poco confiable se da a entender que no hay garantías para que un datagrama alcance su destino. Sin embargo, proporciona ciertos mecanismos de aviso de errores, cuando algo va mal, por ejemplo que algún ruteador tenga su buffer lleno, IP elimina el datagrama y trata de enviar un mensaje de ICMP de vuelta al origen de la transmisión del datagrama. Cualquier tipo de mecanismo de confiabilidad deberá ser implementado por la capa superior, por ejemplo TCP.

Aunque IP proporciona servicios connectionless, en general la capa de red puede proporcionar servicios de dos tipos: Los orientados a conexión y los no orientados a conexión.

Si los servicios son no orientados a conexión (connectionless) como es el caso de IP, o sea, en base a datagramas, la capa de red transmite hacia la red real un paquete de datos, que incluye la dirección a la que deberá enviarse. La red deberá hacer su mayor esfuerzo para enviar los datos, pero hay probabilidades que los datos se pierdan, se dañen o sean duplicados.

Así mismo, cada paquete tendrá una ruta diferente, y la red no garantiza que estos llegarán en orden a su destino. El modelo connectionless es similar al servicio postal.

Si los servicios son orientados a conexión (en base a circuitos virtuales), se tienen tres fases en la transmisión de datos entre dos entidades de la capa de red, en la primera se establece la conexión, después se pueden intercambiar datos, y al final se libera la conexión.

Esto es similar al sistema telefónico. El sistema telefónico es informado por el usuario del destino que desea alcanzar (discando el número), después de esto, se establece un camino entre el usuario que llama y el teléfono que se disco, generalmente reservando recursos de cierto tipo, que permanecerán asignados por toda la duración de la llamada, después de que la llamada es aceptada, se puede conversar. Un servicio orientado a conexión tiene las siguientes características:

La red garantiza que todos los paquetes serán enviados en orden, sin perdida o duplicación, si ocurre algo que no lo permita, la red desconectara a las entidades en comunicación.

Se establece un camino único para el flujo de datos, y todos los datos fluyen a través de él, si algo sucede que no permita seguir este camino, la red desconecta a las entidades en comunicación.

La red garantiza cierta capacidad de transmisión (ancho de banda) para las entidades que se comunican, si no la utilizan, el recurso se desperdicia.

Si la red se satura, se rechaza cualquier intento de establecer una conexión (como en las compañías telefónicas).

Existen algunas discrepancias acerca de cual de los dos tipos de servicios es mejor, si el de orientado a conexión que el no orientado a conexión. Los que defienden los servicios orientados a conexión señalan:

Las redes son utilizadas de manera primordial para transferir archivos y para tener acceso a sistemas remotos a través de la emulación de terminal. Estas aplicaciones requieren que los paquetes permanezcan ordenados, y no toleran la pérdida de paquetes. Esto es complejo, es mejor que este servicio se encuentre en la red, lejos de los sistemas que desean comunicarse.

Los servicios orientados a conexión, permiten que los ruteadores sean más rápidos, ya que las decisiones complejas se realizan al inicio de la comunicación, después bastará con que el ruteador busque en una tabla el camino que deberá seguir el paquete de datos.

Es mejor rechazar conexiones que afectar la capacidad de transmisión sobre las que ya están establecidas.

Es mejor para la capa de transporte, que la capa de red defina un camino único para toda la conexión, ya que así la capa de transporte podrá determinar el tamaño del segmento que sea mas eficiente. También será más sencillo determinar el tamaño de segmento que sea más eficiente, también será más sencillo determinar los niveles de retransmisión que deberán adoptarse por la pérdida de acuse de recibo ya que se conocerá el tiempo de viaje de cada paquete.

Por otra parte, aquellos que defienden los servicios connectionless argumentan:

La mayoría de las redes orientadas a conexión, están construidas de tal manera que si algo falla durante la transmisión, el circuito se interrumpe inmediatamente, tal vez, sin posibilidad de que se recuperen los paquetes que en ese momento se encontraban en transito. Por lo tanto, en vez de evitar esfuerzos para la capa de transporte, los duplica.

Muchas aplicaciones, no requieren que los paquetes sean enviados en orden. En particular, la transmisión de voz, en donde un pequeño porcentaje de paquetes perdidos no afecta la calidad del sonido.

También la transferencia de archivos puede ser implantada sobre la base de un protocolo no secuencial, que sería mas adecuado para redes de alta velocidad.

El trafico en una red no es uniforme, es decir, hay momentos en los que se transmiten datos de manera masiva, y hay otros en los que no existe nada. Es mejor tener una división estadística de los recursos que apartar para que se desperdicien.

ISO nunca llego a algún acuerdo sobre que esquema utilizar, y se adoptaron los dos, para los servicios orientados a conexión, generalmente se utiliza X.25 de CCITT, y para los servicios sin conexión se utiliza CLNP ISO 8473. Dentro de la arquitectura de TCP/IP, IP es un protocolo conectionless.

La implantación del servicio de red connectionless es mas complicado que el orientado a circuitos virtuales. Como un ejemplo de esto podemos mencionar el caso de la fragmentación de paquetes (el tamaño máximo de un datagrama esta en función al medio de transmisión que este siendo utilizado). En el caso de X.25, simplemente se marca cada paquete con un bit M el cual indicaba que seguían mas paquetes. En cambio en un servicio orientado a datagramas, esto no es suficiente, ya que los paquetes no se reciben en orden. También deberá pensarse en el mecanismo que la capa de red deberá seguir para ensamblar estos paquetes, lo cual es costoso en recursos de memoria y procesador.

Conceptualmente, una red TCP/IP proporciona tres conjuntos de servicios, tal como lo muestra la figura 2.1, este arreglo nos sugiere algún tipo de dependencia entre los servicios. En el nivel inferior, un servicio de envíos orientado a conexión proporciona la base sobre la cual los otros servicios se apoyan. En el siguiente nivel, una capa de transporte confiable proporciona una plataforma de nivel mas alto de la cual la capa de aplicación depende.

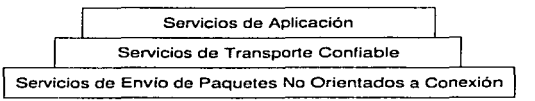


Figura 2.1. Las tres capas conceptuales de los servicios de Internet.
Nótese que entre cada una de estas capas, existe una interdependencia notable.

El sistema más fundamental de Internet es el sistema de envío de paquetes. Técnicamente, el servicio es definido como un sistema de envío de paquetes poco confiable y no orientado a conexión.

Los paquetes se pueden duplicar, perder, retrasar o ser enviados en desorden, sin embargo, este sistema no detecta este tipo de problemas y no informa a las partes en comunicación de dichos eventos.

Es servicio es llamado no orientado a conexión debido a que cada paquete es tratado en forma individual, cada paquete puede viajar por distintas rutas o perderse.

El protocolo que define los mecanismos poco confiables y no orientados a conexión es el llamado protocolo de Internet (IP). IP proporciona tres importantes definiciones. Primero, IP define las unidades básicas de transferencia de datos usados a través de una Internet TCP/IP. Esto es, especifica el formato exacto de todos los datos que pasan a través de la Internet TCP/IP. Segundo, el Software de IP realiza una función de ruteo, escogiendo la ruta que seguirán los datos. Tercero, además de precisar las especificaciones formales de los formatos de datos y el ruteo, IP incluye una serie de reglas que engloban la idea del envío de paquetes poco confiable. Las reglas se caracterizan como los hosts y gateways deben de procesar los paquetes, como y cuando se deben de generar los mensajes de error y las condiciones para que un paquete sea descartado.

Se puede hacer una analogía entre una red física y una red TCP/IP. En una red física, la unidad de transferencia es un frame, el cual contienen un encabezado y datos, donde el encabezado da información tal como la dirección física origen y destino.

Las redes TCP/IP llaman a su unidad básica de transferencia un datagrama IP. Tal como en las redes físicas, un datagrama se divide en un encabezado y un área de datos. También en los datagramas se tiene una dirección origen y una dirección destino, la cual va contenida dentro del encabezado. La diferencia radica en que el frame contiene direcciones físicas y el datagrama contiene direcciones IP.

La figura 2.2 muestra el formato de un datagrama IP. En este datagrama, el bit menos significativo es numerado como 0 a la izquierda, y el más significativo de los 32 bits es numerado como 31 en el lado derecho.

Los cuatro bytes son transmitidos en el siguiente orden: Primero los bits 0-7, después del 8-15, 16-23 y al final del 24-31. Esto se llama ordenamiento de bytes *big endian*, el cual es el orden requerido para todos los enteros binarios en los encabezados de TCP/IP que son enviados y transpuestos para la red. Esto se conoce como orden de bytes de red. Las máquinas que almacenan enteros binarios en otros formatos, tales como el formato *little endian*, deben de convertir los valores de los encabezados a un orden de byte de red antes de transmitir los datos.

2.3. *Formato del Encabezado de IP*

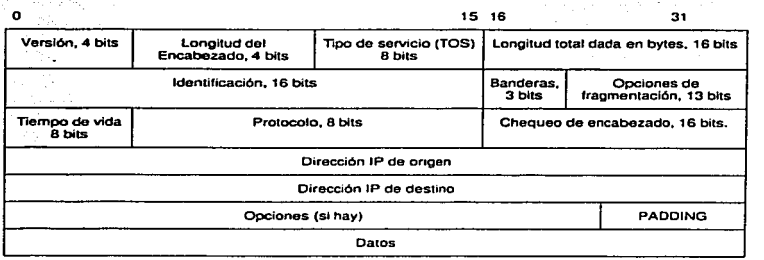


Figura 2.2. Formato del encabezado de IP.
El datagrama es la unidad básica de transferencia en una Internet TCP/IP

Dado que el procesamiento del datagrama ocurre en el software, el contenido y formato del mismo no es relacionado con ningún hardware. Por ejemplo, los primeros 4 bits del datagrama contienen la versión de protocolo IP que fue utilizado para crear el datagrama. Se utiliza para verificar que tanto la máquina que envía el datagrama, la que lo recibe y los gateways en medio de ellos estén de acuerdo en el formato del datagrama. Cualquier software IP debe de checar este campo antes de procesar el datagrama, para corroborar que el datagrama es compatible con el formato que espera. Si la versión es diferente, la máquina rechazará el datagrama. Actualmente se utiliza IP versión 4 (Ipv4) para la transmisión de datos, sin embargo, actualmente se realizan pruebas de la nueva versión de IP llamada Ipv6, el cual será abordado más adelante.

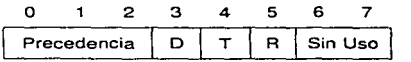
El campo de longitud del encabezado tienen una longitud de 4 bits y proporciona el tamaño del encabezado IP medido en palabras de 32 bits. El campo de longitud total nos proporciona el tamaño de todo el datagrama IP, esta longitud se mide en bytes. Como el campo de longitud total es de sólo 16 bits, el tamaño máximo que puede utilizarse es de 65535 octetos.

Por el momento, parece ser suficiente este tamaño, sin embargo para redes de mayor velocidad, el tamaño de datagrama se verá limitado por esta característica.

2.3.1. El campo de Tipo de Servicio

El tipo de servicio es usado para indicar la calidad de servicio deseado. El tipo de servicio es un conjunto abstracto o generalizado de atributos, los cuales caracterizan los servicios disponibles que proporcionan las redes que conforman la Internet. Esta indicación de tipo de servicio es usada por los gateways para seleccionar los parámetros actuales de transmisión de una red, la red a ser usada en el siguiente salto o al siguiente gateway cuando se rutea un datagrama IP.

El campo de tipo de servicio tiene la siguiente estructura:



Valores de la sección de Precedencia
111 = Control de Red
110 = Control de la Internet
101 = CEITIC/ECP
100 = Flash Override
011 = Flash
010 = Inmediata
001 = Prioridad
000 = Rutina

Figura 2.3. Estructura del campo de Tipo de Servicio.

Los tres bits del campo de precedencia especifican, como su nombre lo indica, la precedencia del datagrama, puede tener valores de 0 (precedencia Normal) a 7 (control de red), permitiendo a las entidades en comunicación indicar la importancia de cada datagrama. Aunque la mayoría de los hosts y gateways ignoran el tipo de servicio, es un concepto importante porque proporciona un mecanismo que eventualmente permitirá el control de información para tener precedencia sobre los datos. Por ejemplo, si todos los hosts y gateways tuvieran el precedencia máxima, es posible implementar algoritmos de control congestión que no son afectados por la congestión que están tratando de controlar.

Los bits D, T y R especifican el tipo de transporte que se desea para el datagrama. Cuando se pone en 1 el bit D, se pide un retardo bajo, el bit T especifica la máxima transferencia posible y el bit R realiza la indicación de alta confiabilidad en la transmisión. Sin embargo, puede ser posible que la red no pueda proporcionar este tipo de servicio, sin embargo, esto no indica que el datagrama se descarte, sino que se continua con la transmisión. A continuación muestro una tabla en donde se pueden apreciar los valores mas comunes para algunas de las aplicaciones utilizadas en Internet.

Protocolo	TOS	Valor
TELNET	1000	Minimizar Retardo
FTP		
Control	1000	Minimizar Retardo
Datos	0100	Maximizar tasa de Transferencia
TFTP	1000	Minimizar Retardo
SMTP		
Fase de Comandos	1000	Minimizar Retardo
Fase de datos	0100	Maximizar tasa de Transferencia
Domain Name Service		
Petición UDP	1000	Minimizar Retardo
Petición TCP	0000	
Zone Transfer	0100	Maximizar tasa de Transferencia
NNTP	0001	Minimizar costo Monetario
ICMP		
Errores	0000	
Peticiones	0000	
Respuestas	Igual que peticiones	
Cualquier IGP	0010	Maximizar Confiabilidad
EGP	0000	
SNMP	0010	Maximizar Confiabilidad
BOOTP	0000	

Figura 2.4. Valores comunes para el ruteo de datagramas de algunas aplicaciones en Internet.

2.3.2. Fragmentación de Datagramas

Antes de continuar con los siguientes campos del datagrama de IP, es necesario considerar como los datagramas son enviados a la red física. El primer punto a considerar es el máximo tamaño que puede alcanzar un datagrama. Los datagramas son manejados por el software, sin embargo, los frames están intrínsecamente ligados al hardware. Los datagramas pueden ser del tamaño que el diseñador elija, sin embargo, para la versión 4 de IP, hemos visto que el tamaño máximo de la longitud total del datagrama es de 65,535 octetos, pero este valor puede aumentarse para las nuevas versiones de IP.

La mayoría de las limitaciones en el tamaño del datagrama aparecen en la práctica. Sabemos que los datagramas se mueven de máquina a máquina transportados por la capa física. Para hacer este transporte eficiente, deseamos garantizar que cada datagrama viaja en un frame físico distinto. Esto es, deseamos que nuestra abstracción de un paquete de red sea mapeado directamente a un paquete real si es posible.

La idea de llevar un datagrama en un frame de red es llamado encapsulación. Para las capas inferiores un datagrama IP es como cualquier otro mensaje enviado de una máquina a otra. El Hardware no reconoce el formato del datagrama ni la dirección IP destino.

En el caso ideal, el datagrama IP encaja perfectamente bien a un frame físico, haciendo la transmisión a través de la red física eficiente. Para obtener tal eficiencia, los diseñadores de IP podían haber seleccionado un tamaño máximo de datagrama en donde tal datagrama podría siempre encajar en un frame. Sin embargo el problema es el frame elegido para realizar esto, porque los datagramas pueden pasar a través de diferentes tipos de redes físicas cada una de las cuales puede tener un tamaño de frame distinto.

Para entender el problema, necesitamos tomar en cuenta un hecho acerca del hardware de red: cada tecnología de switcheo de paquetes tiene un límite superior con respecto a la cantidad de datos que pueden ser transferidos en un frame de capa física. Por ejemplo, el límite de transferencia de Ethernet es de 1500 octetos de datos, mientras que pro-NET permite 2044 octetos en un frame. Estos valores son conocidos como MTU (Maximum Transfer Units, Unidades Máximas de Transferencia). Los tamaños de MTU pueden ser realmente pequeños, algunas tecnologías limitan la transferencia a 128 octetos o menos. Limitando a los datagramas a corresponder el MTU mínimo posible en Internet hace que la transferencia sea ineficiente cuando estos datagramas pasan a una red que puede manejar frames de tamaño superior.

Como se ha de recordar, uno de los objetivos de esta capa es ocultar los detalles de las capas inferiores y hacer la comunicación conveniente para el usuario. TCP/IP elige un tamaño de datagrama conveniente y proporciona mecanismos para dividir datagramas grandes en piezas mas pequeñas cuando los datagramas tiene que pasar por redes con un MTU pequeño. Las piezas pequeñas en las cuales un datagrama es dividido es denominado fragmento y el proceso para dividir un datagrama es llamado fragmentación.

Cada uno de los fragmentos del datagrama debe tener un tamaño cercano al MTU, pero debe de conservarse el criterio de que el tamaño debe ser un múltiplo de 8 octetos. Por esta razón, algunos de los fragmentos tendrán tamaños distintos. Cada uno de los fragmentos tiene un formato similar al del datagrama original, ya que casi toda la información se repite, a excepción del campo de banderas, el cual indica que el datagrama se ha fragmentado.

Sin embargo, aquí surge la duda si se debe de reensamblar el datagrama o se deben de enviar cada una de las piezas al destino final. En TCP/IP, una vez que un datagrama ha sido fragmentado, los fragmentos viajan como un datagrama separado y es al final del recorrido cuando se reensamblan.

Preservar los fragmentos hasta el destino tiene dos desventajas: primero, dado que los datagramas no son reensamblados inmediatamente, después de pasar una red con un MTU pequeño, los fragmentos pequeños deben de ser enviados desde el punto de fragmentación hasta el destino final, esto trae consigo algo de ineficiencia, debido a que, aunque se encuentren redes con un MTU grande, no se podrá aprovechar y pasarán sólo piezas pequeñas de información. Segundo, si alguno de los fragmentos se pierde, todo el datagrama será inservible.

La máquina receptora inicia un reloj de reensamble cuando llega el primer fragmento. Si el reloj expira antes de que todos los fragmentos lleguen, la máquina receptora descarta las piezas sin procesar el datagrama. Por lo tanto, la probabilidad de pérdida de datagramas aumenta cuando se realiza fragmentación.

Pasando por alto las desventajas, realizar el re-ensamble de datagramas en el destino final trabaja bien. Permite a los fragmentos ser ruteados en forma independiente y no requiere que los ruteadores intermedios almacenen o reensamblen fragmentos.

Existen tres campos en el encabezado del datagrama que controlan la fragmentación y reensamble de los datagramas: el campo de identificación, las banderas y opciones de fragmentación. El campo de identificación contiene un entero único que identifica al datagrama, cada fragmento del datagrama debe de contener el mismo número de datagrama en su encabezado. Su propósito primario es permitir al destino distinguir de que datagrama es el fragmento está llegando. Conforme llega cada fragmento, el destino utiliza el campo de identificación junto con la dirección origen del datagrama para identificarlo. Las computadoras que envían los datagramas IP deben generar un valor único para el campo de identificación por cada datagrama. Hay una técnica utilizada por el software IP que establece un contador global en memoria, el cual se incrementa cada vez que se crea un datagrama nuevo y asigna el resultado al campo de identificación del datagrama.

Hay que recordar que cada fragmento tiene exactamente el mismo formato que un datagrama completo. Para un fragmento, el campo de opciones de fragmentación especifica el desplazamiento en el datagrama original de los datos que se están acarreando en el fragmento, medido en unidades de 8 octetos, comenzando con un desplazamiento igual a cero. Para reensamblar el datagrama, el destino debe obtener todos los fragmentos comenzando con el fragmento que tiene asignado un desplazamiento igual a 0 hasta el fragmento con el desplazamiento de mayor valor. Los fragmentos no necesariamente llegaran en orden, además no hay comunicación entre el ruteador que fragmentó el datagrama y el destino que trata de reensamblarlo.

Los 2 bits de orden menor del campo de 3 bits de banderas, controlan la fragmentación. Por lo general, el software de aplicación que utiliza TCP/IP no se ocupa de la fragmentación debido a que tanto la fragmentación y el reensamble son procedimientos automáticos que se dan a bajo nivel en el sistema operativo,

invisible para el usuario final. Sin embargo, para probar el software de red o depurar problemas operacionales, podría ser importante probar el tamaño de los datagramas en los que se presenta la fragmentación. El primer bit de control ayuda a esta prueba especificando en que momento se debe fragmentar un datagrama. Se le conoce como bit de no fragmentación cuando esta puesto a 1 significa que el datagrama no debe fragmentarse. Una aplicación podría no permitir la fragmentación cuando solo el datagrama completo es útil.

Por ejemplo, consideremos la secuencia de iniciación de una computadora, en la que una máquina comienza a ejecutar un pequeño programa en ROM y utiliza la red para solicitar una primera inicialización, y otra máquina envía de regreso una imagen de memoria. Si el software ha sido diseñado así, necesitará la imagen completa, pues de otra forma no le será útil; por ello, el datagrama debe tener activado el bit de no fragmentación. Cada vez que un ruteador necesita fragmentar un datagrama que tiene activado el bit de no fragmentación, el ruteador descartará el datagrama y devolverá un mensaje de error a la fuente.

El bit de orden inferior en el campo de FLAGS especifica si el fragmento contienen datos intermedios del datagrama original o de la parte final. Este campo es conocido como more fragments (mas fragmentos). Para entender por que este bit es necesario, consideremos el software IP en el destino final cuando trata de reensamblar un datagrama. Este recibirá los fragmentos (es posible que en desorden) y necesitará saber cuando ha recibido todos los fragmentos de un datagrama. Cuando un fragmento llega, el campo de longitud total en el encabezado indica el tamaño del fragmento y no el tamaño total del datagrama; por esta razón el destino no puede utilizar el campo de longitud total para determinar si ha reunido todos los fragmentos. El bit de mas fragmentos resuelve este problema con facilidad: cada vez que, en el destino se recibe un fragmento con el bit mas fragmentos desactivado, se sabe que este fragmento acarrea datos del extremo final del datagrama original. De los campos opciones de fragmentación y longitud total se puede calcular la longitud del datagrama original.

2.3.3. Tiempo de Vida del datagrama.

El campo de tiempo de vida especifica la duración, en segundos, del tiempo que el datagrama tiene permitido permanecer en la red. La idea es sencilla e importante: cada vez que una máquina introduce un datagrama dentro de la red, se establece un tiempo máximo durante el cual el datagrama puede permanecer ahí. Los ruteadores y hosts que procesan los datagramas deben decrementar el campo de tiempo de vida cada vez que pasa un datagrama y eliminarlo de la red cuando su tiempo ha concluido.

Es difícil determinar este parámetro, ya que los ruteadores por lo general no conocen el tiempo de tránsito por las redes físicas. Unas pocas reglas simplifican el procedimiento y hacen fácil el manejo de datagramas sin relojes sincronizados.

En primer lugar, cada ruteador a lo largo de un trayecto, desde una fuente hasta un destino, es configurado para decrementar en 1 el campo de tiempo de vida cuando se procesa el encabezado del datagrama. Sin embargo, para manejar casos de ruteadores sobrecargados que introducen retardos largos, cada ruteador registra el tiempo local cuando llega un datagrama y decrementa el tiempo de vida por el número de segundos que el datagrama permanece dentro del ruteador esperando a que se le despache.

Una vez el campo de tiempo de vida alcanza el valor de cero, el ruteador descarta el datagrama y envía un mensaje de error al origen . La idea de establecer un temporizador para los datagramas es interesante, ya que garantiza que los datagramas no viajarán por la red indefinidamente, aun cuando las tablas de ruteo se corrompa y los ruteadores direccionen datagramas en un ciclo.

El campo de chequeo de encabezado asegura la integridad de los valores del encabezado. Este procedimiento se lleva a cabo considerando al encabezado como una secuencia de enteros de 16 bits (en el orden de los octetos de la red), sumándolos juntos mediante el complemento a uno, y después tomando el complemento a uno del resultado. Para propósitos de cálculo del chequeo de encabezado se toma este campo como cero.

2.3.4. Las opciones del Datagrama

Las opciones proporcionan las funciones necesarias o usadas en algunos casos pero innecesarias para comunicaciones comunes. Las opciones se incluyen en principio para pruebas de red o depuración. Sin embargo el procesamiento de las opciones es parte integral del protocolo IP, por lo tanto, todos los estándares de implementaciones se deben incluir.

La longitud del campo de opciones varia dependiendo de que opción sea seleccionada. Algunas opciones tienen una longitud de un solo octeto. Cuando las opciones están presentes en un datagrama apareces contiguas, sin separadores especiales entre ellas. Cada opción consiste de un solo octeto de código de opción que debe llevar a continuación un solo octeto y un conjunto de octetos de datos para cada opción. El octeto de código de opción se divide en tres campos como se muestra en la siguiente figura:

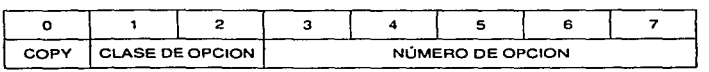


Figura 2.5 División del octeto de código de opción en tres campos de 1, 2 y 5 bits.

El campo consiste de una bandera de 1 bit llamada COPY, un segmento de 2 bits llamado clase de opción y un segmento de 5 bits llamado número de opción. La bandera COPY controla la forma en que los ruteadores tratan las opciones

durante la fragmentación. Cuando el bit COPY esta puesto a 1, especifica que la opción se debe copiar en todos los fragmentos. Cuando esta puesto a cero el bit COPY significa que la opción sólo se debe copiar dentro del primer fragmento y no en todos los fragmentos.

Los bits de clase de opción y número de opción especifican la clase general de opción y establecen una opción específica en esta clase. La tabla de la figura 2.6 muestra como se asignan las clases.

La tabla de la figura muestra las opciones posibles que pueden acompañar a un datagrama IP y muestra los valores para las clases de opción y número de opción.

Clase de Opción	Número de Opción	Longitud	Descripción
0	0	-	Fin de la lista de opciones. Se utiliza si las opciones no terminan al final del encabezado (ver también campo de relleno de encabezado)
0	1	-	Sin operación se utiliza para alinear octetos en una lista de opciones)
0	2	11	Seguridad y restricciones de manejo (para aplicaciones militares)
0	3	Variable	Ruteo no estricto de origen. Se utiliza para rutear un datagrama a través de una trayectoria específica.
0	7	Variable	Registro de la ruta. Se utiliza para registrar el trayecto de una ruta.
0	8	4	Identificador de flujo. Se utiliza para transportar un identificador de flujo SATNET (obsoleto)
0	9	Variable	Ruteo estricto de origen. Se utiliza para establecer la ruta de un datagrama en un trayecto específico.
2	4	Variable	Sello de tiempo Internet. Se usa para registrar sellos de hora a lo largo de una ruta.

Figura 2.6 . Opciones posibles del datagrama IP.
La mayor parte de las opciones se utiliza con propósito de control.

2.3.5. IP en el contexto de TCP/IP

IP no proporciona una herramienta de comunicación confiable. No hay avisos de llegadas. No hay control de error de datos, solo un chequeo del encabezado. No hay retransmisiones. No hay flujo de control.

Los errores pueden ser reportados por medio del ICMP (Internet Control Message Protocol) el cual es implementado en el modulo de IP.

IP esta específicamente limitado a proporcionar las funciones necesarias para enviar un datagrama de una fuente a un destino sobre un sistema interconectado de redes. No hay mecanismos para argumentar la confiabilidad de transmisión entre los destinatarios, control de flujo, secuencia o otros servicios comúnmente encontrados en protocolos host-host. IP puede capitalizar los servicios de las redes que soporta para proporcionar varios tipos de calidad de servicio.

Este protocolo es llamado por protocolos host-host en un ambiente Internet. Este protocolo llama a su vez a otros protocolos de red para llevar los datagramas al siguiente gateway o al host destino.

Por ejemplo, un modulo TCP puede llamar un modulo de IP para tomar un segmento TCP (incluyendo el encabezado y los datos de usuario) como una porción de un datagrama de IP. El modulo de TCP puede proporcionar la dirección y otros parámetros en el encabezado de IP al modulo de IP como argumento de la llamada. El modulo de Internet puede entonces crear un datagrama IP y llamar en la interfaz de red para transmitir el datagrama IP.

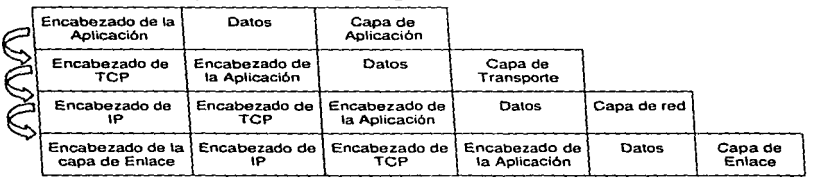


Figura 2.7. Esquema crecimiento en el tamaño del paquete de datos a través de las capas.
Cada capa del esquema de TCP/IP agrega un encabezado de control al paquete de datos que le envía la capa superior para transmisión. Cuando se recibe el paquete, cada capa elimina el encabezado que agregó y le pasa a la capa superior el paquete "limpio".

2.4. Modo de Operación.

Para poder entender mas a fondo que características y como son proporcionados los servicios de la capa de red, deberemos analizar la estructura de direccionamiento y las características que deben tener las direcciones en este nivel.

En la vida diaria, estamos acostumbrados a utilizar direcciones jerárquicas, por ejemplo, cuando enviamos una carta, especificamos el país, la ciudad, la colonia, la ciudad, la calle y el número dentro de esa calle al cual la carta deberá llegar, esto es una estructura jerárquica.

La complejidad de un sistema de asignación de nombres de calles que garantizara un nombre único a nivel mundial sería inmensa. Por lo anterior, al especificar en que país, ciudad y colonia se encuentra una calle, reducimos el problema a que el responsable de los nombres de calles dentro de una colonia asigne nombres únicos.

Con las redes de computadoras sucede algo similar. En primer lugar, las direcciones deberán ser únicas en la región de influencia de la red, si hablamos de Internet, esto es a nivel mundial. Por otra parte, el trabajo de un centro que designara direcciones únicas sería enorme. Por eso se designan centros regionales que tienen la capacidad de definir una parte de la dirección jerárquica.

Si definimos cuatro niveles en la estructura jerárquica, una dirección ejemplo se vería como 100.140.53.200, en donde cada uno de los puntos representa una jerarquía. Cabe mencionar por ejemplo, que las direcciones de la capa de enlace no son jerárquicas, ya que esta capa solo tienen por objeto suministrar conectividad entre puntos contiguos, típicamente estos puntos contiguos existen dentro de la misma subred.

Conceptualmente cada dirección IP es un par (netid, hostid), en donde netid identifica a la dirección de una red y hostid identifica una máuina dentro de la red. Las direcciones IP fueron diseñadas para distinguir rápidamente los campos nerid y hostid. Los ruteadores que utilizan el campo netid de una dirección para decidir a donde enviar un paquete, dependen de una extracción eficiente para lograr una velocidad alta. La codificación de información de red en una dirección de Internet tiene algunas desventajas. La desventaja mas obvia es que las direcciones se refieren a las conexiones de red, no a la computadora en sí, por lo tanto, si la máquina se mueve de lugar, su dirección IP se debe cambiar.

La distribución de las direcciones de red y de nodo pueden variar según el tipo de red de la que sea. Sin embargo, esto puede ajustarse mas a las necesidades del administrador de la red a través del mecanismo de máscaras. Una mascara ajustará dentro de la parte del host, que parte representará al host realmente, y que parte representará las estructuras jerárquicas internas que esa red pueda tener.

La aplicación de la mascara se hace de la siguiente manera: se toma la dirección destino y se hace una operación de AND lógica para "ocultar" la parte de la dirección del host, una vez que queda solamente la parte de la dirección de red, se compara con las tablas de ruteo para encontrar así el lugar por donde será enviado el paquete. La siguiente figura muestra como se realiza la operación con la mascara.

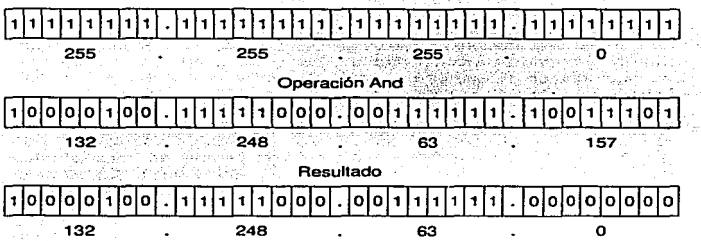


Figura 2.8 Utilización de la mascara de red.
La operación de AND da como resultado la separación de la dirección de red de la de host.

Existen números con significado especial cuando se utilizan dentro de una dirección, por ejemplo, el número 0 representa "este" y el número 1 "todos". Por ejemplo, la dirección 132.248.255.255 es un broadcast hacia la red 132.248, la 132.248.63.255 es un broadcast hacia la subred 132.248.63, la dirección 0.0.0.0 se utiliza cuando el host no conoce su dirección y desea solicitarla. Otra dirección especial es la 127.0.0.1 que representa un loopback.

Además existen las direcciones clase D o de multicast, es decir, especifican la dirección de mas de una máquina, estas se distinguen porque comienzan con 1110.

Los módulos de IP usan campos en el encabezado para fragmentar y ensamblar los datagramas IP cuando es necesario transmitirlos a través de redes de paquetes pequeños.

El modelo de operación es que un modulo IP reside en cada host checando la comunicación en cada ruteador que interconecta las redes. Esos módulos comparten reglas comunes para interpretar los campos de dirección y para fragmentar y ensamblar datagramas IP. Además esos módulos (especialmente en gateways) tienen procedimientos para hacer decisiones de ruteo y otras funciones.

El protocolo IP maneja cada datagrama como una entidad independiente sin relación con cualquier otro datagrama. No hay conexiones o circuitos lógicos.

En un sistema de conmutación de paquetes, el ruteo es el proceso de selección de un camino sobre el que se mandarán paquetes y el ruteador es la computadora que hace la selección.

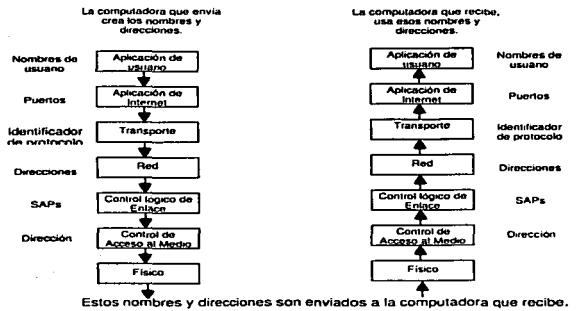
El ruteo ocurre a muchos niveles. Por ejemplo, dentro de una red de área amplia que tiene muchas conexiones físicamente entre conmutadores de datos, la red por sí misma es responsable de rutear paquetes desde que llegan hasta que salen. Dicho ruteo interno está completamente contenido dentro de la re de área amplia. Las máquinas en el exterior no pueden participar en las decisiones; sólo ven la red como una entidad que entrega paquetes.

Recordemos que el objetivo de IP es proporcionar una red virtual que comprenda muchas redes físicas, así como ofrecer un servicio son conexión de entrega de paquetes. De forma análoga al ruteo dentro de una red física, el ruteo IP selecciona un camino por el que se debe enviar un datagrama. El algoritmo de ruteo debe escoger como enviar un datagrama pasando por muchas redes físicas.

El ruteo en una Internet puede ser difícil, en especial entre computadoras que tienen muchas conexiones físicas de red. De forma ideal, el software de ruteo examina aspectos como la carga de la red, la longitud del datagrama o el tipo de servicio que se especifica dentro del datagrama para seleccionar el mejor camino. Sin embargo, la mayor parte del software de ruteo en Internet es mucho menos sofisticado y selecciona rutas basándose en suposiciones sobre los caminos mas cortos.

Existen varios tipos de direcciones: direcciones físicas, de enlace de datos y en la capa de red, sin embargo, estas direcciones son insuficientes para mover los paquetes a su destino final en la máquina destino. Por ejemplo, un paquete puede ser destinado a una aplicación específica, tal como al correo electrónico o algún sistema de transferencia de archivos. Dado que estas dos aplicaciones residen en la misma capa superior (la capa de aplicación), se debe implementar un mecanismo para que el paquete de datos llegue en forma correcta a la aplicación deseada.

Los nombres y direcciones de las capas superiores son identificados por una amplia variedad de términos. Una convención para Internet es usar los términos de identificador de protocolo, puerto y socket. La convención del Modelo de Referencias OSI establece el término de service access point (SAP).



Dirección MAC Destino	DSAP	Dirección IP Destino	Identificador de Protocolo	Puerto Destino	Nombre de Usuario Destino	Datos de Usuario
Dirección MAC Origen	SSAP	Dirección IP Origen	Identificador de Protocolo	Puerto Origen	Nombre de Usuario Origen	Datos de Usuario

Figura 2.9 Relación de Nombres y Direcciones en las capas de Internet

La figura anterior muestra la relación de las direcciones entre las diferentes capas dentro de Internet.

En la capa de Red se tienen direcciones IP, sin embargo, para conocer el mejor camino que debe recorrer un paquete de datos es necesario contar con algoritmos de descubrimientos de rutas eficientes, los cuales permitan tener una visión mas clara de la red.

2.5. Algoritmos de Ruteo

Existen diferentes algoritmos para el descubrimiento de rutas, los dos mas utilizados son el Bellman-Ford o Vector de distancias y el de Costo de Enlace.

2.5.1. Algoritmo de Vector de Distancia

A continuación mostraré el algoritmo de vector de distancias utilizando una analogía. Una persona se encuentra en un poblado que también es una intersección de caminos, su trabajo es colocar avisos en esta intersección, estos avisos deberán contener el nombre de cada poblado y la distancia que falta para llegar a el. Puede comenzar colocando un letrero con el nombre de esta intersección indicando que faltan 0 Km. Entonces, puede continuar midiendo la distancia que existe entre este poblado y sus vecinos inmediatos, anotando además la información que se encuentra en los letreros de ese poblado. Después

de un rato, y si en cada poblado había una persona dedicada a la misma labor, contará con un conjunto de letreros que indicarán las distancias a cada poblado, y la dirección en la que se encuentra (aquella en donde hayamos visto que suma la distancia mas corta). Esto se ilustra en la siguiente figura:

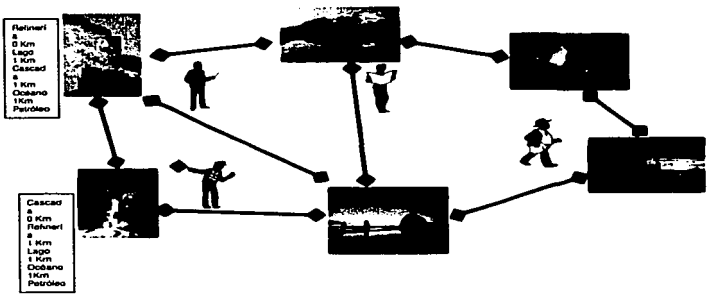


Figura 2.10 Ejemplo de el Algoritmo de Vector de Distancias

El algoritmo para una red de computadoras va como sigue:

- Cada ruteador es configurado con su propio ID
- Cada ruteador es configurado, de tal manera que a cada enlace le corresponda un número, o costo de enlace.
- Cada ruteador, comienza con un vector de distancias de valor 0 para el mismo, y un valor de infinito para cualquier otro destino.
- Cada ruteador transmite su vector de distancias a cada vecino, siempre que la información cambie.
- Cada ruteador salva el vector de distancias mas reciente que haya recibido de cada uno de sus vecinos.
- Cada ruteador calcula su propio vector de distancias, minimizando el costo de cada destino, a través del examen del costo hacia ese destino que haya sido reportado por cada vecino, agregándole el costo de enlace hacia ese vecino.

Los siguientes eventos causan un recálculo del vector de distancias:

- Recibir de un vecino, un vector de distancias diferente.

➢ Descubrir que el enlace a un vecino se ha caído.

En este algoritmo, el ruteador establece una lista de todas las rutas conocidas en una tabla. Cuando arranca, un ruteador inicia esta tabla de ruteo para que contenga una entrada de información por cada red conectada directamente. Cada introducción en la red identifica una red destino y establece una distancia hacia la red, por lo general medida en saltos.

Aún cuando los algoritmos de vector de distancia son fáciles de implementar, tienen desventajas. En un ambiente completamente estático, los algoritmos de vector de distancia difunden rutas hacia todos los destinos. Cuando las rutas cambian rápidamente, sin embargo, los cómputos podrían no ser estables. Cuando una ruta cambia (por ejemplo si aparece una nueva conexión o si una conexión vieja falla), la información se propaga lentamente de un ruteador a otro. Esto significa que algunos ruteadores pueden tener información de ruteo incorrecta.

2.5.2. Algoritmo de Estado de Enlaces

La principal desventaja de los algoritmos de vector de distancia es que no se extienden bien. Junto con el problema de respuesta lenta a cambios en la configuración de la red, el algoritmo requiere de intercambio de mensajes largos.

Dado que la actualización de los mensajes de ruteo contienen una entrada de información para cada red posible, el tamaño de los mensajes es proporcional al número total de redes en una Internet.

Además, debido a que un protocolo de vector de distancias requiere de la participación de todos los ruteadores, el volumen de información a intercambiar puede ser enorme.

La principal alternativa a los algoritmos de vector de distancia es una clase de algoritmos conocidos como enlace-estado. Los algoritmos de este tipo requieren que cada ruteador participante tenga información de la topología completa. La forma mas sencilla de pensar la información de la topología es imaginando que todos los ruteadores tienen un mapa que muestra a todos los otros ruteadores y las redes a las que están conectados. En términos abstractos, los ruteadores corresponden a los nodos o vértices en un grafo y las redes que conectan a los ruteadores corresponden a los arcos. Hay un arco entre dos nodos si y solo si los correspondientes ruteadores pueden conectarse directamente.

En lugar de enviar un mensaje qu contenga una lista de destinos, un ruteador que participa en un algoritmo de este tipo desempeña dos tareas. En primer lugar, prueba activamente el estado de todos los ruteadores vecinos. En términos de un grafo, dos ruteadores son vecinos si comparten un enlace; en términos de una red, dos vecinos están conectados a una red común. En segundo lugar, difunde periódicamente la información del estado de enlace hacia los otros ruteadores.

Para probar el estado de un vecino conectado directamente, un ruteador intercambia de manera periódica mensajes cortos que interrogan si el vecino esta

conectado y activo. Si el vecino responde, se dice que el enlace esta levantado, de otra forma se dice que el enlace esta caído.

2.5.3. Ruteo de Datagramas IP

El algoritmo usual de ruteo IP emplea una tabla de ruteo Internet (a veces, conocida como tabla de ruteo IP) en cada máquina que almacena información sobre posibles destinos y sobre como alcanzarlos. Debido a que tanto los ruteadores como los hosts rutean datagramas, ambos tienen tablas de ruteo IP. Siempre que el software de ruteo IP en un host necesita transmitir un datagrama, consulta la tabla de ruteo para decidir a donde enviarlo.

Surge ahora el cuestionamiento de la información que deberá albergar la tabla de ruteo. Si cada tabla de ruteo contuviera información sobre cada posible dirección de destino, sería imposible mantener actualizada las tablas. Además, como el número de destinos posibles es muy grande, las máquinas no tendrían suficiente espacio para almacenar la información.

De manera conceptual, nos gustaría utilizar el principio de ocultación de información y permitir a las máquinas tomar decisiones de ruteo con información mínima. Por ejemplo, nos gustaría aislar la información sobre hosts específicos del ambiente local en el que existen y hacer que las máquinas que estén lejos ruteen paquetes hacia ellos sin saber dichos detalles. Por fortuna, el esquema de direcciones IP nos ayuda a lograr este objetivo.

Hay que recordar que las direcciones IP se asignan de tal manera que todas las máquinas conectadas a una red física comparten un prefijo en común (la porción de red en la dirección). Por esta razón las tablas de ruteo sólo necesitan contener prefijos de red y no direcciones IP completas.

2.5.3.1. Ruteo con salto al siguiente nodo

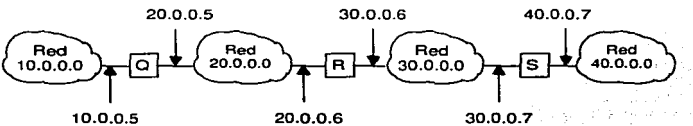
Por lo común, una tabla de ruteo contiene pares (N,R), donde N es la dirección IP de una red destino y R es la dirección IP del siguiente ruteador en el camino hacia la red N. El ruteador R es conocido como el salto siguiente y la idea de utilizar una tabla de ruteo para almacenar un salto siguiente para cada destino es conocida como ruteo con salto al siguiente. Por lo tanto, la tabla de ruteo en el ruteador R sólo especifica un paso a lo largo del camino de R a su red siguiente. Por lo tanto, la tabla de ruteo en el ruteador R sólo especifica un paso a lo largo del camino de R a su red destino — el ruteador no conoce el camino completo hacia el destino.

Es importante entender que cada registro en una tabla de ruteo apunta hacia un ruteador que se puede alcanzar a través de una sola red. Esto es, que todos los ruteadores listados en la tabla de ruteo de la máquina M deben residir en las redes con las que M se conecta de manera directa. Cuando un datagrama está listo para dejar M, el software IP localiza la dirección IP de destino y extrae la

porción de red. Luego, M utiliza la porción de red para tomar una decisión de ruteo, seleccionando un ruteador que se pueda alcanzar directamente.

El la práctica, también aplicamos el principio de ocultación de información a los anfitriones. Insistimos que, aunque los anfitriones tengan tablas de ruteo IP, deben guardar información mínima en ellas. La idea es obligar a los anfitriones a que deleguen la mayor parte de sus funciones de ruteo a los ruteadores.

En la figura 2.11, se muestra un ejemplo concreto que nos ayuda a explicar las tablas de ruteo. La red ejemplificada consiste en cuatro redes conectadas por tres ruteadores. En la figura la tabla de ruteo proporciona las rutas que utiliza el ruteador R. Ya que R se conecta de manera directa a 20.0.0.0 y 30.0.0.0, puede utilizar la entrega directa para llevar a cabo un envío a un host en cualquiera de esas redes (posiblemente utilizando ARP para encontrar las direcciones físicas). Teniendo un datagrama destinado para un anfitrión en la red 40.0.0.0, R lo rutea a la dirección 30.0.0.7 que es la dirección del ruteador S. Luego, S entregará el datagrama en forma directa. R puede alcanzar la dirección 30.0.0.7 debido a que tanto R como S se conectan de manera directa con la red 30.0.0.0.



(a)

Para alcanzar los hosts en la red	Rutear a esta dirección
20.0.0.0	Entregar directamente
30.0.0.0	Entregar directamente
10.0.0.0	20.0.0.5
40.0.0.0	30.0.0.7

(b)

Figura 2.11 (a) Ejemplo de una red con 4 redes y 3 ruteadores, y (b) la tabla de ruteo de R

Como se muestra en la figura 2.11, el tamaño de la tabla de ruteo depende del numero de redes en la red; solamente crece cuando se agregan nuevas redes. Sin embargo, el tamaño y contenido de la tabla son independiente del número de hosts individuales conectados a las redes.

Escoger rutas basándose tan solo en el identificador de red destino tiene muchas consecuencias. Primero, en la mayor parte de los desarrollos, significa que todo el tráfico destinado a una cierta red toma el mismo camino. Como resultado, aun cuando existan muchos caminos, quizá no se utilicen constantemente. De igual manera, todos los tipos de tráfico siguen el mismo camino sin importar el retraso o la generación de salida de las redes físicas. Segundo, debido a que sólo el ruteador del camino intenta comunicarse con el host final, solamente el ruteador puede determinar si el host existe o está en operación. Por lo tanto, necesitamos encontrar una forma para que envíe reportes sobre problemas de entrega, de vuelta a la fuente original. Tercero, debido a que cada ruteador rutea el tráfico en forma independiente, los datagramas que viajan de un host A a B pueden seguir un camino totalmente distinto al que siguen los datagramas que viajan del host B a A.

Necesitamos asegurarnos de que los ruteadores cooperen para garantizar que siempre sea posible la comunicación bidireccional.

2.5.3.2. Rutas asignadas por omisión.

Otra técnica utilizada para ocultar información y mantener reducido el tamaño de las tablas de ruteo, es asociar muchos registros a un ruteador asignado por omisión. La idea es hacer que el software de ruteo IP busque primero la tabla de ruteo de la red destino. Si no aparece una ruta en la tabla, las rutinas de ruteo envían el datagrama a un ruteador asignado por omisión.

El ruteo asignado por omisión es de gran ayuda cuando un sitio tiene pocas direcciones locales y solo una conexión con el resto de Internet. Por ejemplo, las rutas asignadas por omisión trabajan bien en hosts que se conectan a una sola red física y alcanzan sólo un ruteador, el cual es la ruta de acceso al resto de la Internet. Toda la decisión de ruteo consiste en dos comprobaciones: una de la red local, y un valor asignado por omisión que apunta hacia el único ruteador posible. Inclusive si el sitio sólo contiene unas cuantas redes locales, el ruteo es sencillo ya que consiste en pocas comprobaciones de las redes locales, mas un valor asignado por omisión para todos los demás destinos.

2.1.1.1. Rutas por host específico

Aunque he dicho que todo el ruteo esta basado en redes y no en hosts individuales, la mayor parte del ruteo IP permite que se especifiquen rutas por hosts como caso especial. Tener rutas por host le da al administrador de red local un mayor control sobre el uso de la red, le permite hacer comprobaciones y también se puede utilizar para controlar el acceso por razones de seguridad. Cuando se depuran conexiones de red o tablas de ruteo, la capacidad para especificar una ruta especial hacia una máquina individual resulta ser especialmente útil.

2.1.1.2. Algoritmo de ruteo IP

Tomando en cuenta todo lo anteriormente dicho, el algoritmo de ruteo IP es como sigue:

- Al llegar un datagrama, se extrae la dirección IP destino y calcular el prefijo de red.
- Si el prefijo de la red corresponde a cualquier dirección de red directamente conectada entregar el datagrama al destino sobre esa red. (Esto comprende la transformación de D en una dirección física, encapsulando el datagrama y envío del frame).
- En caso contrario, si la tabla contiene una ruta con host especifico, enviar el datagrama al salto siguiente especificado en la tabla de ruteo.
- De otra forma, si la tabla contiene una ruta para una red N, enviar el datagrama al salto siguiente especificado en la tabla.
- De otra forma, si la tabla contiene una ruta asignada por omisión, enviar el datagrama al ruteador asignado por omisión especificado en la tabla de ruteo.
- En caso contrario, enviar un error de ruteo.

Es importante entender que, a excepción de la disminución del tiempo de vida y de volver a calcular el checksum, ruteo IP no altera el datagrama original. En particular, las direcciones de origen y destino del datagrama permanecen sin alteración; estas siempre especifican la dirección IP fuente y destino original. Cuando IP ejecuta el algoritmo de ruteo, selecciona una nueva dirección IP, que es la dirección IP de la máquina a la que a continuación se tendrá que enviar el datagrama. La nueva dirección es parecida a la dirección de un ruteador. Sin embargo, si el datagrama se puede entregar directamente, la nueva dirección será la misma que la del último destino.

2.6. ICMP

En el apartado anterior se mostró cómo el software de IP proporciona un servicio de entrega de datagramas, no confiable y sin conexión, la hacer que cada ruteador direccione datagramas. Un datagrama viaja de ruteador en ruteador hasta que llega a uno que lo puede entregar directamente a su destino final. Si un ruteador no puede entregar o rutear un datagrama, o si el ruteador detecta una condición anormal que afecta su capacidad para direccionarlo (por ejemplo, congestionamiento de red), necesita informar a la fuente original para que evite o corrija el problema. Aquí se analizaran los mecanismos que utilizan los ruteadores y hosts de Internet para comunicar la información de control o error.

En el sistema no orientado a conexión que proporciona IP, cada ruteador opera de manera autónoma, ruteando o entregando los datagramas que llegan sin coordinarse con el transmisor original. El sistema trabaja bien si todas las máquinas funcionan de manera correcta y si están de acuerdo respecto a las rutas. Por desgracia, ningún sistema funciona bien todo el tiempo.

Además de las fallas en las líneas de comunicación y en los procesadores, IP tiene fallas en la entrega de datagramas cuando la máquina destino está desconectada temporal o permanentemente de la red, cuando el contador de tiempo de vida expira, o cuando los ruteadores intermedios se congestionan tanto que no pueden procesar el tráfico entrante. La mas importante diferencia entre tener una sola red implantada con hardware dedicado y tener una red implantada con software es que, en el primer caso, el diseñador puede añadir hardware especial para informar a los hosts conectados cuando surge un problema.

En Internet no se tiene un mecanismo de hardware como el anterior, un transmisor no puede indicar si ocurrió una falla en la entrega, originada por un mal funcionamiento local o un o remoto. La depuración se vuelve muy difícil. El protocolo IP, por lo mismo no contiene nada para ayudar al transmisor a comprobar la conectividad ni para ayudarle a aprender sobre dichas fallas.

Para permitir que los ruteadores en una red reporten los errores o proporcionen información sobre circunstancias inesperadas, los diseñadores agregaron a los protocolos TCP/IP un mecanismo de mensajes de propósito especial.

El mecanismo conocido como ICMP (Internet Control Message Protocol, Protocolo de Mensajes de Control de Internet), se considera como parte obligatoria de IP y se debe incluir en todas las implantaciones IP.

Al igual que el resto del tráfico, los mensajes de ICMP viajan a través de Internet en la porción de datos de IP. Sin embargo, el destino final de un mensaje de ICMP no es un programa de aplicación ni un usuario en máquina destino, sino el software de IP de dicha máquina. Esto es, cuando llega un mensaje de error ICMP, el módulo de software ICMP lo maneja. Por su puesto, si el ICMP determina que un protocolo de un nivel superior o un programa de aplicación causaron problemas, notificará al módulo apropiado.

Aunque ICMP fue diseñado para permitir que los ruteadores reporten a los hosts las causas de errores en la entrega, ICMP no se restringe sólo a los ruteadores. Aunque las reglas y normas limitan el uso de algunos mensajes ICMP, cualquier máquina puede enviar un mensaje ICMP a cualquier otra. Por lo tanto, un host puede utilizar ICMP para comunicarse con un ruteador o con otro host. La mayor ventaja de permitir que los host utilicen ICMP es que proporciona un solo mecanismo que se utiliza para todos los mensajes de información y control.

Técnicamente ICMP es un mecanismo de reporte de errores. Proporciona una forma para que los ruteadores que encuentran un error lo reporten a la fuente original. Aunque la especificación del protocolo subraya los usos deseables de ICMP y sugiere acciones posibles para responder a los reportes de error, ICMP no especifica del todo la acción que debe tomarse para cada posible error.

La mayor parte de los errores provienen de la fuente origina, pero otros no. Sin embargo, debido a que ICMP reporta los problemas a la fuente original, no se puede utilizar para informar los problemas a los ruteadores intermedios.

Los mensajes de ICMP requieren dos niveles de encapsulación, como se muestra en la figura 2.12. Cada mensaje ICMP viaja a través de Internet en la porción de datos de IP, el cual viaja a través de cada red física en la porción de datos de un frame.

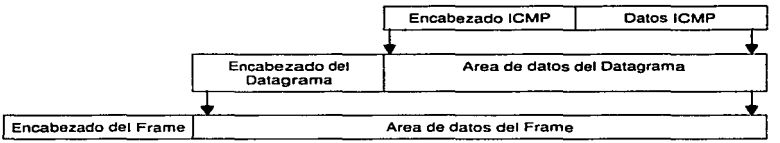


Figura 2.12. Niveles de encapsulación de ICMP
El mensaje ICMP se encapsula en un datagrama IP que, a su vez se encapsula en un frame para su transmisión. Para identificar a ICMP, el campo protocolo del datagrama contiene el valor de 1 (valor asignado a ICMP).

Los datagramas que llevan mensajes ICMP se rutean exactamente como los que llevan información de usuario; no existe ni una confiabilidad ni una prioridad adicionales. Por lo tanto, los mensajes pueden perderse o descartarse. Además, en una red congestionada, el mensaje de error puede congestionar aun mas la red. Hay una excepción en los procedimientos de manejo de errores si un datagrama IP que lleva un mensaje ICMP causa un error. Esta excepción esta diseñada para evitar el problema de tener mensajes de error sobre mensajes de error, especifica que los mensajes ICMP no se generan por errores resultantes de datagramas que llevan mensajes de error ICMP.

Es importante tener en mente que aunque los mensajes de ICMP se encapsulan y envían mediante IP, ICMP no se considera como un protocolo de un nivel mas alto sino como una parte obligatoria de IP. La razón de utilizar IP para entregar mensajes de ICMP es que quizá necesiten viajar a través de muchas redes físicas para alcanzar su destino final. Por lo tanto no se pueden entregar sólo por medio de transporte físico.

2.6.1. Formato de los mensajes de ICMP

Aunque cada mensaje ICMP tiene su propio formato, todos comienzan con los mismos tres campos; un campo de tipo de mensaje (8 bits y números enteros)

que identifica el mensaje; un campo de código (8 bits) que proporciona mas información que proporciona mas información sobre el tipo de mensaje y un campo de Checksum (16 bits) . Además, los mensajes ICMP que reportan errores siempre incluyen el encabezado y los primeros 64 bits de datos del datagrama que causó el problema.

La razón de regresar más que el encabezado del datagrama únicamente es para permitir que el receptor determine de manera mas precisa que protocolo(s) y programa de aplicación son responsables del datagrama. Como veremos mas adelante, los protocolos de mas alto nivel de la suite de TCP/IP están diseñados para codificar la información crucial en los primeros 64 bits.

El campo Tipo de ICMP define el significado del mensaje así como su formato. Los tipos incluyen:

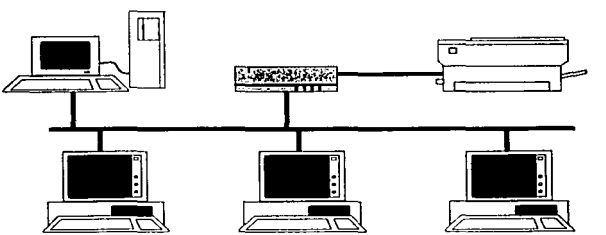
Valor del campo	Tipo del mensaje
0	Respuesta de Eco
3	Destino inaccesible
4	Disminución de origen
5	Redireccionar (cambiar una ruta)
8	Solicitud de eco
11	Tiempo excedido para un datagrama
12	Problema de parámetros en un datagrama
13	Solicitud de TimeStamp
14	Respuesta de TimeStamp
15	Solicitud de información (obsoleto)
16	Respuesta de Información
17	Solicitud de mascara de dirección
18	Respuesta de mascara de dirección

Una de las herramientas de depuración mas utilizadas incluye los mensajes ICMP de petición y repuesta de eco. En muchos sistemas, el comando que llama el usuario para enviar solicitudes de eco ICMP se conoce como ping. Las versiones mas sofisticadas de ping envían una serie de solicitudes de eco ICMP, capturan las respuestas y proporcionan estadísticas sobre la pérdida de datagramas. Permiten que el usuario especifique la longitud de los datos que se envían, así como el intervalo entre solicitudes.

- Esquemas de Protocolos por capas
- La idea del Multiplexado y Demultiplexado
- UDP
- TCP
- Dependencias entre protocolos

3

La capa de Transporte (TCP)



Lista de Figuras

En los capítulos anteriores he revisado los fundamentos de la arquitectura del enlace de redes, describí como los ruteadores y host transmiten datagramas en Internet y presenté los mecanismos utilizados para asociar direcciones IP a direcciones de la red física. También describí como TCP/IP es capaz de transferir datagramas IP entre hosts, donde cada datagrama es ruteado a través de la red, basándose en la dirección IP de destino.

En IP, una dirección destino identifica un host; no se hace ninguna distinción con respecto a que usuario o que programa de aplicación recibirá el datagrama. En este capitulo se amplía el grupo de protocolos TCP/IP al agregar un mecanismo que distingue entre muchos destinos dentro de un host, permitiendo que otros programas de aplicación que se ejecutan en una computadora envíen y reciban datagramas forma independiente.

3.1.Esquema de protocolos por capas.

Los protocolos permiten especificar o entender una forma de comunicación sin conocer los detalles de hardware de red de un vendedor en particular. Los sistemas complejos de comunicación de datos no utilizan un solo protocolo para manejar todas las tareas de transmisión, sino que requieren de un conjunto de protocolos cooperativos, a veces llamados familia de protocolos o suite de protocolos. Para entender por qué, pensemos en los problemas que se pueden presentar cuando las máquinas se comunican a través de una red de datos:

- Fallas en el Hardware. Un host o ruteador puede fallar, ya sea porque el hardware falle o porque el sistema operativo quede fuera de servicio. Un enlace de transmisión de red puede fallar o desconectarse accidentalmente. El software de protocolo necesita detectar estas fallas y restablecer el funcionamiento.
- Congestionamiento en la red. Aun cuando el hardware y el software funcionen correctamente, estos tienen una capacidad finita que puede ser excedida. El software de protocolo debe implantar un arreglo en las vías de transmisión para que una máquina congestionada no entorpezca el tráfico.
- Paquetes retrasados o perdidos. Algunas veces, el envío de paquetes tiene retrasos muy largos o éstos se pierden. El software de protocolo necesita aprender acerca de las fallas o debe adaptarse a los retardos.
- Corrupción de datos. La interferencia eléctrica, magnética o las fallas en el hardware pueden ocasionar errores de transmisión que alteran el contenido de los datos transmitidos. El software de protocolo necesita detectar y repara estos errores.
- Errores en la secuencia de los datos o duplicación de datos. Las redes que ofrecen múltiples rutas pueden entregar los datos fuera de secuencia o entregar paquetes duplicados. El software de protocolo necesita reordenar los paquetes y suprimir los duplicados.

Si se consideran en conjunto, todos estos problemas parecen abrumadores. Es difícil entender como se podría escribir un solo protocolo para manejar todos estos problemas. Por ello, se utilizan diferentes protocolos, los cuales atacan cada uno problemas específicos. Sin embargo, se debe considerar ciertos aspectos en el diseño de una estructura por capas. Primero se debe realizar una estandarización de los formatos de mensajes que intercambiaran los protocolos. Después se debe establecer la secuencia de operación de cada uno de estos protocolos.

Pensemos en los módulos de software de protocolo en una máquina como una pila vertical constituida por capas, tal como se muestra en la siguiente figura:

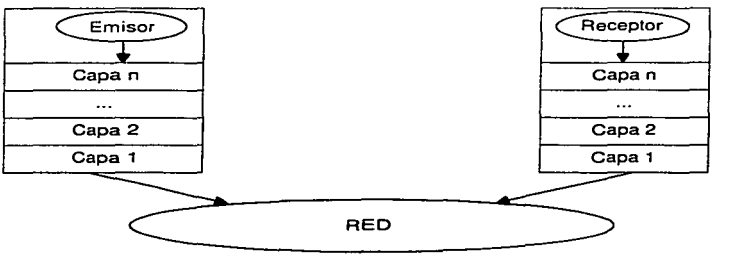


Figura 3.1 Organización conceptual del software de protocolo en capas

Conceptualmente, enviar un mensaje desde un programa de aplicación en una máquina hacia un programa de aplicación en otra, significa transferir el mensaje hacia abajo, por las capas sucesivas del software de protocolo en la maquina emisora, transferir el mensaje a través de la red, y luego transferir el mensaje hacia arriba, a través de las capas sucesivas del software de protocolo en la máquina receptora.

En la práctica, el software de protocolo es mucho más complejo de lo que se muestra en el modelo simplificado de la figura 3.1. Cada capa toma decisiones acerca de lo correcto del mensaje y selecciona una acción apropiada con base en el tipo de mensaje a través de la red y, selecciona una acción apropiada con base en el tipo de mensaje o la dirección destino. Por ejemplo, una capa en la máquina de recepción debe decidir cuando tomar un mensaje o enviarlo a otra máquina. Otra capa debe decidir que programa de aplicación deberá recibir el mensaje.

Para entender la diferencia entre la organización conceptual del software de protocolo y los detalles de implantación, consideremos la comparación que se muestra en la figura 3.2. El diagrama conceptual en la figura 3.2a muestra una capa de Internet entre una capa de protocolo de alto nivel y una capa de interfaz de red. El diagrama realista de la figura 3.2 b muestra el hecho de que el software IP puede comunicarse con varios módulos de protocolo de alto nivel y con varias interfaces de red.

Aun cuando un diagrama conceptual de la estratificación por capas no muestra todos los detalles, sirve como ayuda para explicar los conceptos generales.

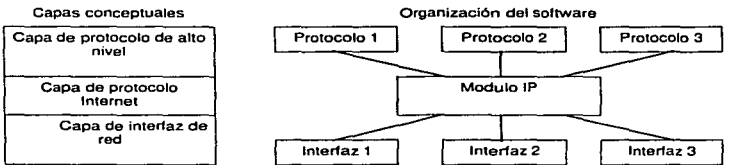


Figura 3.2 Una comparación de (a) estratificación por capas conceptual de protocolos y, (b) visión realista de la organización del software que muestra varias interfaces de red entre IP y varios protocolos.

Una vez que se ha tomado la decisión de subdividir los problemas de comunicación en subproblemas y organizar el software de protocolos en módulos, de manera que cada uno maneje un subproblema, surge la pregunta: " ¿qué funciones deberán implementarse en cada módulo?" La respuesta no es fácil de responder por varias razones. En primer lugar, un conjunto de objetivos y condiciones determinan un problema de comunicación en particular, es posible elegir una organización que optimice el software de protocolo para ese problema. Segundo, incluso cuando se consideran los servicios generales en el ámbito de red, como un transporte confiable, es posible seleccionar entre distintas maneras de resolver el problema. Tercero, el diseño de una arquitectura de red y la organización del software de protocolo están interrelacionados; no se puede diseñar a uno sin considerar al otro.

Existen dos ideas dominantes sobre la estratificación por capas de protocolos. La primera, basada en el trabajo realizado por la ISO, conocido como modelo de referencia OSI, y el conjunto de protocolos TCP/IP.

En términos generales, el software TCP/IP está organizado en cuatro capas conceptuales que se construyen sobre una quinta capa de hardware. La figura 3.3 muestra las capas conceptuales así como la forma en que los datos pasan entre ellas.

Independientemente del esquema de estratificación por capas que se utilice o de las funciones de las capas, la operación de los protocolos estratificados por capas se basa en una idea fundamenta.

La idea conocida como principio de estratificación por capas se basa en protocolos diseñados de modo que una capa n en el receptor de destino reciba exactamente el mismo objeto enviado por la correspondiente capa n de la fuente.

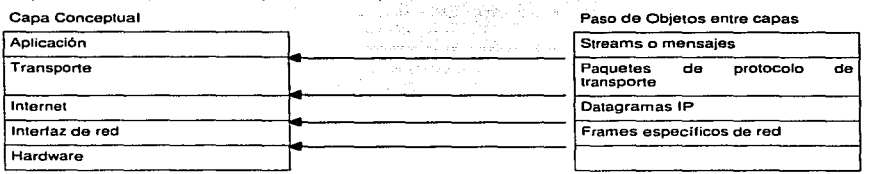


Figura 3.3 Las cuatro capas conceptuales del software TCP/IP y la forma en que los objetos pasan entre ellas.

La capa con el nombre de interfaz de red se conoce con frecuencia con el nombre de capa de enlace de datos.

El principio de estratificación por capas explica por qué la estratificación por capas es una idea poderosa. Esta permite que el diseñador de protocolos enfoque su atención hacia una capa a la vez, sin preocuparse acerca del desempeño de las capas inferiores.

La estratificación por capas incluye dos fronteras que podrían no ser obvias; una frontera de dirección de protocolo que separa los direccionamientos de alto nivel y de bajo nivel, una frontera de sistema operativo que separa al sistema de los programas de aplicación. El principio de estratificación podría resumirse con la siguiente figura:

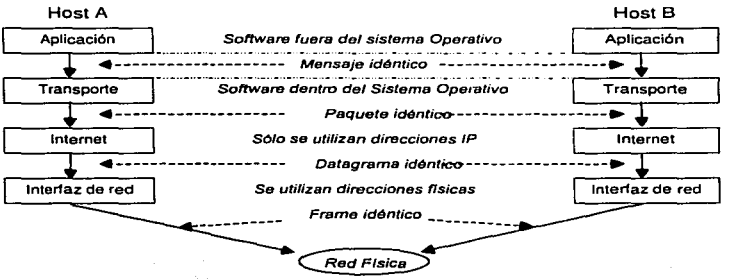


Figura 3.4. Principio de estratificación.

Aquí se puede observar que el host B recibe exactamente lo que el host A le envía en cada una de las capas.

Ahora que conocemos el funcionamiento del modelo por capas, podemos profundizar en conceptos tales como multiplexado y demultiplexado. Estos conceptos están presentes en los modelos de referencia, tales como OSI y el mismo TCP/IP.

3.2.La idea del multiplexado y demultiplexado

Los protocolos de comunicación utilizan técnicas de multiplexado y demultiplexado a través de la jerarquía de capas. Cuando envía un mensaje, la computadora fuente incluye bits extras que codifican el tipo de mensaje, el programa de origen y los protocolos utilizados. Finalmente, todos los mensajes son colocados dentro de frames de red para transferirse y combinarse en streams de paquetes. En el extremo de recepción, la máquina destino se vale de la información extra para guiar el proceso.

Consideremos el ejemplo de demultiplexado que se muestra en la figura 3.5

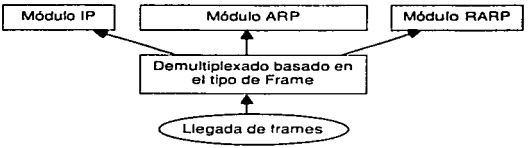


Figura 3.5. Demultiplexado de tramas entrantes basado en el campo de tipo que se encuentra en el encabezado del frame.

La figura muestra la forma en que el software utiliza en la capa de interfaz de red el tipo de frame para seleccionar un procedimiento que permita manejar las tramas entrantes. Se dice que la interfaz de red demultiplexa el frame en base a su tipo. Para hacer posible la selección, el software en la máquina origen debe establecer el campo del tipo de frame antes de la transmisión. Así cada módulo de software que envía frames emplea el campo de tipo para especificar el contenido del frame.

El multiplexado y demultiplexado se presentan en casi todas las capas de protocolo. Por ejemplo, luego que la interfaz de red demultiplexa los frames y pasa los frames que contienen datagramas IP hacia el módulo IP, el software IP extrae el datagrama y lo demultiplexa con base en el protocolo de transporte.

Para decidir como manejar un datagrama, el software de IP examina el encabezado de un datagrama, y para su manejo, selecciona un protocolo con base en el tipo de datagrama. Por ejemplo, puede ser ICMP, UDP o TCP entre otros.

Los sistemas operativos de la mayor parte de las computadoras aceptan la multiprogramación, esto significa permitir que varios programas de aplicación se ejecuten al mismo tiempo.

Utilizando la jerga de los sistemas operativos, nos referimos a cada programa en ejecución como un proceso, tarea, programa de aplicación o proceso en el ámbito de usuario; estos sistemas son conocidos como sistemas multitarea.

Puede parecer natural decir que un proceso es el destino final de un mensaje. Sin embargo, especificar que un proceso en particular en una maquina en particular es el destino final para un datagrama es un poco confuso. Primero, por que los procesos se crean y destruyen de manera dinámica, los transmisores rara vez tienen suficiente información para identificar un proceso en otra maquina. Segundo, sería deseable poder reemplazar los procesos que reciben datagramas, sin tener que informar a todos los transmisores (por ejemplo, reiniciar una máquina puede cambiar todos los procesos, pero los transmisores no están obligados a saber sobre los nuevos procesos). Tercero, necesitamos identificar Los destinos de las funciones que implantan sin conocer el proceso que implanta la función (por ejemplo, permitir que un transmisor contacte un servidor de archivos sin saber que, proceso en la máquina de destino implanta la función de servidor de archivos). También es importante saber que, en los sistemas que permiten que un solo proceso maneje dos o más funciones, es esencial que encontremos una forma para que un proceso decida exactamente qué función desea el transmisor.

En vez de pensar en un proceso como destino final, imaginaremos que cada máquina contiene un grupo de puntos abstractos de destino, llamados puertos de protocolo. Cada puerto de protocolo se identifica por medio de un número entero positivo. El sistema operativo local proporciona un mecanismo de interfaz que los procesos utilizan para especificar o accesar un puerto.

La mayor parte de los sistemas operativos proporciona un acceso síncrono a Los puertos. Desde el punto de vista de un proceso en particular, el acceso síncrono significa que las operaciones se detienen durante el acceso a puerto. Por ejemplo, si un proceso intenta extraer datos de un puerto antes de que llegue cualquier dato, el sistema operativo detiene (bloquea) temporalmente el proceso hasta que lleguen datos. Una vez que esto sucede, el sistema operativo pasa los datos al proceso y lo vuelve a iniciar. En general, Los puertos tienen buffers, para que los datos que llegan antes de que un proceso esté listo para aceptarlos no se pierdan. Para lograr la colocación en buffers, el software de protocolo, localizado dentro del sistema operativo, coloca los paquetes que llegan de un puerto de protocolo en particular en una cola de espera (finita) hasta que un proceso los extraiga.

Para comunicarse con un puerto externo, un transmisor necesita saber tanto la dirección IP de la maquina de destino como el número de puerto de protocolo del destino dentro de la máquina. Cada mensaje debe llevar el número del puerto de destino de la máquina a la que se envía, así como el número de puerto de origen de la máquina fuente a la que se deben direccionar las respuestas. Por lo tanto, es posible que cualquier proceso que recibe un mensaje conteste al transmisor.

3.3. UDP

En el grupo de protocolos TCP/IP, el Protocolo de Datagrama de usuario o UDP proporciona el mecanismo primario que utilizan los programas de aplicación para enviar datagramas a otros programas de aplicación. UDP proporciona puertos de protocolo que son utilizados para distinguir entre muchos programas que se ejecutan en la misma máquina. Esto es, además de los datos, cada mensaje UDP contiene tanto el número de puerto de destino como el número de puerto de origen, haciendo posible que el software UDP en el destino entregue el mensaje al receptor correcto y que este envíe una respuesta.

UDP utiliza al Protocolo de Internet subyacente para transportar un mensaje de una máquina a otra y proporciona la misma semántica de entrega de datagramas, sin conexión y no confiable que IP. No emplea acuses de recibo para asegurarse de que llegan los mensajes, no ordena los mensajes entrantes, ni proporciona retroalimentación para controlar la velocidad a la que fluye la información entre las máquinas. Por lo tanto, Los mensajes UDP se pueden perder, duplicar o llegar en desorden, además, los paquetes pueden llegar más rápido de lo que el receptor puede procesarlos.

Un programa de aplicación que utiliza UDP acepta toda la responsabilidad por el manejo de problemas de confiabilidad, incluyendo la pérdida, duplicación y retraso de los mensajes, la entrega en desorden y la pérdida de conectividad. Por desgracia, los programadores de aplicaciones a menudo olvidan estos problemas cuando diseñan software. Además, como los programadores a menudo prueban el software de red utilizando redes de área local, altamente confiables y de baja demora, el procedimiento de pruebas puede no evidenciar las fallas potenciales. Por lo tanto, muchos programas de aplicación que confían en UDP trabajan bien en un ambiente local, pero fallan cuando se utilizan en una red TCP/IP más grande.

3.3.1. Formato del Mensaje UDP

Cada mensaje UDP se conoce como datagrama de usuario. Conceptualmente, un datagrama de usuario consiste de dos partes: un encabezado y un área de datos. Como se muestra en la figura 3.6, el encabezado se divide en cuatro campos de 16 bits, que especifican el puerto desde el que se envió el mensaje, el puerto destino del mensaje, la longitud del mensaje y un campo de checksum.

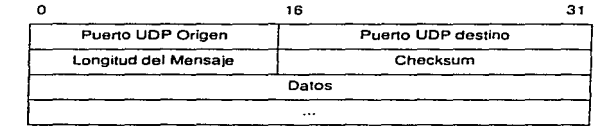


Figura 3.6 Formato de los campos de un datagrama UDP

Los campos de puerto origen y puerto destino contienen los números de puerto del protocolo UDP utilizados para el demultiplexado de datagramas entre los procesos que los esperan recibir. El campo de puerto origen es opcional. Cuando se utiliza, especifica la parte a la que se deben enviar Las respuestas, de lo contrario, puede tener valor cero.

El campo de longitud contiene el numero de octetos en el datagrama UDP incluyendo el encabezado y los datos del usuario UDP. Por lo tanto, el valor mínimo para este campo es ocho, que es la longitud del encabezado.

El checksum de UDP es opcional y no es necesario utilizarlo; un valor de cero en el campo de checksum significa que la suma no se computó. Los diseñadores decidieron hacer opcional la suma de verificación con el fin de permitir que las implantaciones operen con poco trabajo de cómputo cuando se utilice UDP en una red de área local altamente confiable. Sin embargo, hay que recordar que IP no realiza el cálculo del checksum de la porción de datos de un datagrama IP. Así que, la suma de verificación UDP proporciona la única manera de garantizar que los datos lleguen intactos, por lo que se debe utilizar.

Los principiantes, a menudo, se preguntan que sucede con los mensajes UDP en los que el campo de checksum calculado es cero. Un valor calculado de cero es posible debido a que UDP utiliza el mismo algoritmo de checksum que IP: divide los datos en cantidades de 16 bits y computa el complemento a uno de la suma del complemento a uno. De manera sorprendente, el cero no es un problema debido a que la aritmética de los unos tiene dos representaciones para el cero: todos los bits como cero o todos los bits como uno. Cuando la suma de verificación computada es igual a cero, UDP utiliza la representación con todos los bits como uno.

3.3.2. El pseudo-encabezado de UDP

El campo de checksum de UDP abarca más información de la que está presente en el datagrama UDP por sí solo. Para computar la suma de verificación, UDP añade un pseudo-encabezado al datagrama UDP, adjunta un octeto de ceros para rellenar el datagrama y alcanzar exactamente un múltiplo de 16 bits, y computa la suma de verificación sobre todo el conjunto. El octeto utilizado como relleno y el pseudo-encabezado no se transmiten con el datagrama UDP, ni se incluyen en su longitud. Para computar el checksum, el software primero almacena un cero en el campo de checksum, luego, acumula una suma de complemento de 16 bits de todo el conjunto, incluyendo el pseudo-encabezado, el encabezado UDP y los datos del usuario.

El propósito de utilizar un pseudo-encabezado es para verificar que el datagrama UDP llegó a su destino correcto. La clave para entender el uso del pseudo-encabezado reside en darse cuenta de que el destino correcto consiste en una máquina específica y en un puerto de protocolo específico dentro de dicha máquina. Por si mismo, el encabezado UDP sólo especifica el número de puerto de protocolo. Por lo tanto, para verificar un destino, el UDP de la máquina transmisora computa un checksum que cubre tanto la dirección IP de destino como el datagrama UDP. En el destino final, el software UDP revisa en checksum utilizando la dirección IP destino, obtenida del encabezado del datagrama IP que transportó el mensaje UDP. Si la suma concuerda, debe ser verdad que el datagrama llegó al host de destino deseado, así como al puerto de protocolo correcto dentro del host.

El pseudo-encabezado utilizado en el cálculo del checksum consiste de 12 octetos de datos, distribuidos como se muestra en la figura 3.7. Los campos en el pseudo-encabezado etiquetados como dirección IP origen y dirección IP destino contienen las direcciones IP que se utilizarán cuando se envíe el mensaje UDP.

El campo protocolo contiene el código del tipo de protocolo IP (17 para UDP) y el campo de longitud contiene la longitud de datagrama UDP (sin incluir el pseudo-encabezado).

Para revisar el checksum, el receptor debe extraer estos campos del encabezado IP, ensamblarlos en el formato de pseudo-encabezado y volver a calcular la suma.

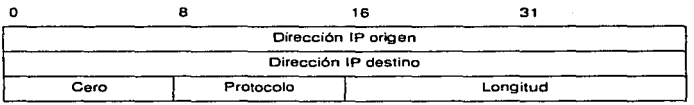


Figura 3.7 Los 12 octetos de un pseudo-encabezado que se utilizan durante el cálculo del checksum de UDP.

UDP proporciona el primer ejemplo de un protocolo de transporte. En el modelo de estratificación por capas, UDP reside sobre la capa del Protocolo Internet. Conceptualmente, los programas de aplicación accesan UDP, que utiliza a IP para enviar y recibir datagramas.

Estratificar por capes UDP por encima de IP significa que un mensaje UDP completo, incluyendo el encabezado UDP y los datos, se encapsula en un datagrama IP mientras viaja a través de Internet, tal como se muestra en la figura 3.8

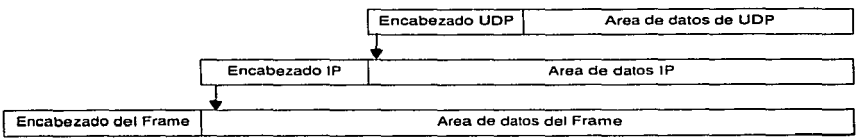


Figura 3.8 Datagrama UDP encapsulado en un datagrama IP para su transmisión a través de Internet.
El datagrama se encapsula en un frame cada vez que viaja a través de una red.

Para los protocolos que hemos examinado, la encapsulación significa que UDP agrega un encabezado a los datos que un usuario envía y lo pasa a IP. La capa IP agrega a su vez un encabezado a lo que recibe de UDP. Y por último, la capa de interfaz de red introduce el datagrama en un frame antes de enviarlo de una máquina a otra. El formato del frame depende de la tecnología subyacente de red. Por lo general, los frames de red incluyen un encabezado adicional.

En la entrada, un paquete llega a la capa más baja del software de red y comienza su ascenso a través de las capas sucesivamente mas altas. Cada capa quita un encabezado antes de pasar el mensaje para que, en el momento en que el nivel mas alto pase los datos al proceso receptor, todos los encabezados hayan sido removidos.

Por lo tanto, el encabezado exterior corresponde a la capa más baja de protocolo y el encabezado interior a la mas alta de protocolo. Cuando se considera cómo se insertan y remueven los encabezados, es importante tener en cuenta el principio de la estratificación por capas. En particular, se debe observar que este principio se aplica a UDP, así que el datagrama UDP que recibió el IP en la máquina de destino es idéntico al datagrama que UDP pasó a IP en la máquina origen. También, los datos que UDP entrega a un proceso usuario en la máquina receptora serán los mismos que un proceso usuario pase a UDP en la máquina transmisora.

La división de funciones entre varias capes de protocolos es inflexible y clara:

La capa IP sólo es responsable de transferir datos entre un par de hosts dentro de Internet, mientras que la capa UDP solamente es responsable de diferenciar entre varias fuentes o destinos dentro de un host.

Por lo tanto, sólo el encabezado IP identifica los hosts de origen y destino; sólo la capa UDP identifica los puertos de origen o destine dentro de un host.

Aparentemente hay una contradicción entre las reglas de la estratificación por capas y el cómputo del checksum en UDP. Recordemos que el checksum incluye un pseudo-encabezado que tiene campos para las direcciones IP de origen y destino. Se puede argumentar que el usuario debe conocer la dirección IP de destino cuando envía un datagrama UDP y que este la debe pasar a la capa de UDP. Por lo tanto, la capa UDP puede obtener la dirección IP de destino sin interactuar con la capa IP. Sin embargo, la dirección IP de origen depende de la ruta que el IP seleccione para el datagrama, debido a que esta dirección identifica la interfaz de red sobre la que se transmite el datagrama. Por lo tanto, UDP no puede conocer una dirección IP de origen a menos que interactue con la capa IP.

Asumimos que el software UDP pide a la capa de IP que compute la dirección IP de origen y (posiblemente) la de destino, las utiliza para construir un pseudo-encabezado, computa la suma de verificación, descarta el pseudo-encabezado y transfiere a la capa IP el datagrama UDP para su transmisión. Con un enfoque alternativo, que produce una mayor eficiencia, se logra que la capa UDP encapsule el datagrama UDP en un datagrama IP, obtenga de IP la dirección de origen, almacene las direcciones tanto de origen como de destino en los campos apropiados del encabezado del datagrama, compute el checksum y

pase el datagrama IP a la capa IP, que sólo necesita llenar los campos restantes del encabezado IP.

¿La fuerte interacción entre UDP e IP viola la premisa básica de que la estratificación por capas refleja la separación de funcionalidad? Si. UDP esta fuertemente integrado al protocolo IP.

Es claramente una transigencia de la separación pura, diseñado enteramente por razones prácticas. Deseamos pasar por alto la violación de estratificación por capas, ya que es imposible identificar plenamente un programa de aplicación de destino sin especificar la maquina de destino y porque queremos realizar, de manera eficaz, la transformación de direcciones utilizadas por el UDP e IP.

3.3.3. Mecanismos de funcionamiento de UDP

Al principio del capítulo, vimos que el software a través de las capas de una jerarquía de protocolos debe multiplexar y demultiplexar muchos objetos en la capa siguiente. El software UDP proporciona otro ejemplo de multiplexado y demultiplexado. Acepta datagramas UDP de muchos programas de aplicación y los pasa a IP para su transmisión, también acepta datagramas entrantes UDP de IP y los transfiere al programa de aplicación apropiado.

Conceptualmente, todo el multiplexado y el demultiplexado entre el software UDP y los programas de aplicación ocurre a través del mecanismo de puerto. En la practica, cada programa de aplicación debe negociar con el sistema operativo para obtener un puerto del protocolo y un número de puerto asociado, antes de poder enviar un datagrama UDP. Una vez que se asigna el puerto, cualquier datagrama que envíe el programa de aplicación a través de el, tendrá el numero de puerto en el campo correspondiente del paquete UDP.

Mientras procesa la entrada, UDP acepta datagramas entrantes del software IP y los demultiplexa, basándose en el puerto de destino UDP

La forma mas fácil de pensar en un puerto UDP es en una cola de espera. En la mayor parte de las implantaciones, cuando un programa de aplicación negocia con el sistema operativo la utilización de cierto puerto, el sistema operativo crea una cola de espera interna que puede almacenar los mensajes que lleguen. A menudo, la aplicación puede especificar o modificar el tamaño de la cola de espera. Cuando UDP recibe un datagrama, verifica si el número de puerto destino corresponde a uno de los puertos que están en uso. Si no, envía mensaje de error ICMP de puerto no accesible y descarta el datagrama. Si encuentra una correspondencia, UDP pone en cola de espera el nuevo datagrama, en el puerto en que lo pueda accesar un programa de aplicación.

¿Cómo se deben asignar los números de puerto de protocolo? El problema es importante ya que dos computadoras necesitan estar de acuerdo en los números de puerto antes de que puedan interoperar. Por ejemplo, cuando la computadora A quiere obtener un archivo de la computadora B, necesita saber que puerto utiliza el programa de transferencia de archivos en la computadora B. Existen dos enfoques fundamentales para la asignación de puertos. El primero se vale de una autoridad central. Todos se ponen de acuerdo en permitir que una autoridad central asigne los números de puerto conforme se necesitan y que

publique la lista de todas las asignaciones. Entonces, todo el software se diseña de acuerdo con la lista. Este enfoque, a veces, se conoce como enfoque universal y las asignaciones de puerto especificadas por la autoridad se conocen como asignaciones bien conocidas de puerto.

El segundo enfoque para la asignación de puertos emplea la transformación dinámica. En este enfoque, los puertos no se conocen de manera global. En vez de eso, siempre que un programa necesita un puerto, el software de red le asigna uno. Para conocer la asignación actual de puerto en otra computadora, es necesario enviar una solicitud que pregunte algo así como "¿que puerto esta utilizando el servicio de transferencia de archivos?" La maquina objetivo responde al proporcionar el número de puerto correcto a utilizar.

Los diseñadores de TCP/IP adoptaron un enfoque híbrido que preasignar algunos números de puerto, pero que deja muchos de ellos disponibles para los sitios locales o programas de aplicación. Los números de puerto asignados comienzan con valores bajos y se extienden hacia arriba, dejando disponibles valores de números enteros altos para la asignación dinámica. En la tabla de la figura 3.9, se listan algunos de los números de puerto UDP actualmente asignados.

Decimal	Palabra Clave	Palabra clave UNIX	Descripción
0	-	-	Reservado
7	ECHO	Echo	Eco
9	DISCARD	Discard	Descartar
11	USERS	Systat	Usuarios activos
13	DAYTIME	Daytime	Hora del día
15	-	Netstat	Quien esta ahí o NETSTAT
17	QUOTE	Qotd	Cita del día
19	CHARGEN	Chargen	Generador de caracteres
37	TIME	Time	Hora
42	NAMESERVER	Name	Servidor de nombres de hosts
43	NICNAME	Whois	Petición de identificación de usuario en NIS
53	DOMAIN	Nameserver	Servidor de nombres de dominios
67	BOOTPS	Bootps	Servidor del protocolo bootstrap
68	BOOTPC	Bootpc	Cliente del protocolo bootstrap
69	TFTP	Tftp	Transferencia trivial de archivos
111	SUNRPC	Sunrpc	RPC de Sun Microsystems

Figura 3.9 Ejemplo ilustrativo de los puertos UDP actualmente asignados, que muestra la palabra clave estándar y su equivalente UNIX; la lista no es completa. En lo posible, otros protocolos de transporte que ofrecen los mismos servicios utilizan los mismos números de puerto que el UDP.

3.4.TCP

Hasta el momento, hemos visto solamente un protocolo de la capa de transporte no orientado a conexión, el cual no permite una transmisión confiable. En esta parte estudiaremos el protocolo TCP (Transport Control Protocol), el cual proporciona servicios orientados a conexión y confiables.

Aunque aquí se presenta a TCP como parte de la suite de protocolos de TCP/IP, es un protocolo independiente de propósitos generales que se puede adaptar para utilizarlo con otros sistemas de entrega. Por ejemplo, debido a que el TCP asume muy poco sobre la red subyacente, es posible utilizarlo tanto en una sola red como Ethernet, como en una Internet compleja. De hecho, TCP es tan popular, que uno de los protocolos para sistemas abiertos de la Organización Internacional para la Estandarización, TP4, se derivo de el.

En el nivel mas bajo, las redes de comunicación por computadora proporcionan una entrega de paquetes no confiable. Los paquetes se pueden perder o destruir cuando los errores de transmisión interfieren con los datos, cuando falla el hardware de red o cuando las redes se sobrecargan demasiado. Las redes rutean dinámicamente los paquetes pueden entregarlos en desorden, con retraso o duplicados. Además, las tecnologías subyacentes de red pueden dictar un tamaño optimo de paquete o formular otras obligaciones necesarias para lograr velocidades eficientes de transmisión.

En el nivel mas alto, los programas de aplicación a menudo necesitan enviar grandes volúmenes de datos de una computadora a otra. Utilizar un sistema de entrega no orientado a conexión y no confiable para las transferencias de grandes volumenes se vuelve tedioso, molesto y requiere que los programadores incorporen, en cada programa de aplicación, la detección y solución de errores. Debido a que es difícil diseñar, entender o modificar el software que proporciona confiabilidad, muy pocos programadores de aplicaciones tienen los antecedentes técnicos necesarios. Como consecuencia, una meta de la investigación de protocolos de red ha sido encontrar soluciones de propósito general para el problema de proporcionar una entrega de flujo confiable, lo que posibilita a los expertos a construir una sola instancia de software de protocolos de flujo que utilicen todos los programas de aplicación. Tener un solo protocolo de propósito general es útil para aislar los programas de aplicación de los detalles del trabajo con redes y permite la definición de una interfaz uniforme pare el servicio de transferencia de flujo.

3.4.1. Servicios confiables

La interfaz entre los programas de aplicación y el servicio TCP/IP de entrega confiable se puede caracterizar por cinco funciones principalmente:

- Orientación de streams. Cuando dos programas de aplicación (procesos de usuario) transfieren grandes volúmenes de datos, pensamos en los datos como un flujo de bits, divididos en octetos de 8 bits, que informalmente se conocen como bytes. El servicio de entrega de flujo en la maquina de destino pasa al receptor exactamente la misma secuencia de octetos que le pasa el transmisor en la maquina de origen.

- Conexión de circuito virtual. La transferencia de flujo es análoga a realizar una llamada telefónica. Antes de poder empezar la transferencia, los programas de aplicación, transmisor y receptor interactuan con sus respectivos sistemas operativos, informándose de la necesidad de realizar una transferencia de flujo. Conceptualmente, una aplicación realiza una "llamada" que la otra tiene que aceptar. Los módulos de software de protocolo en los dos sistemas operativos se comunican al enviarse mensajes a través de Internet, verificando que la transferencia este autorizada y que los dos extremos están listos. Una vez que se establecen todos los detalles, los módulos de protocolo informan a los programas de aplicación que se estableció una conexión y que la transferencia puede comenzar. Durante la transferencia, el software de protocolo en las dos maquinas continua comunicándose para verificar que los datos se reciban correctamente. Si la comunicación no se logra por cualquier motivo (por ejemplo, debido a que falle el hardware de red a lo largo del camino entre las maquinas), ambas maquinas detectaran la falla y la reportaran a los programas apropiados de aplicación. Utilizamos el término circuito virtual pare describir dichas conexiones porque aunque los programas de aplicación visualizan la conexión como un circuito dedicado de hardware, la confiabilidad que se proporciona depende del servicio de entrega de flujo.

- Transferencia con buffers. Los programas de aplicación envían un flujo de datos a través del circuito virtual pasando repetidamente octetos de datos al software de protocolo. Cuando transfieren datos, cada aplicación utiliza piezas del tamaño que encuentre adecuado que puedan ser tan pequeñas como un octeto. En el extremo receptor, el software de protocolo entrega octetos del flujo de datos en el mismo orden en que se enviaron, poniéndolos a disposición del programa de aplicación receptor tan pronto como se reciben y verifican. El software de protocolo puede dividir el flujo en paquetes, independientemente de las piezas que transfiera el programa de aplicación. Para hacer eficiente la transferencia y minimizar el trafico de red, las implantaciones por lo general recolectan datos suficientes de un flujo para llenar un datagrama razonablemente largo antes de transmitirlo a través de una Internet. Por lo tanto, inclusive si el programa de aplicación genera el flujo un octeto a la vez, la transferencia a través de la red puede ser sumamente eficiente. De forma similar, si el programa de aplicación genera bloques de datos muy largos, el software de protocolo puede dividir cada bloque en partes mas pequeñas pare su transmisión.

  Para aplicaciones en las que los datos se deben entregar aunque no se llene un buffer, el servicio de streams proporciona un mecanismo de push que las aplicaciones utilizan para forzar una transferencia. En el extremo transmisor, un push obliga al software de protocolo a transferir todos los datos generados sin tener que esperar a que se llene un buffer. Cuando llega al extremo receptor, el push hace que TCP ponga los datos a disposición de la aplicación sin demora. Sin embargo, se debe hacer notar que la función de push solo garantiza que los datos se transfieran; no proporciona fronteras de registro. Por lo tanto, aun cuando la entrega es forzada, el software de protocolo puede dividir el flujo en formas inesperadas.

- Flujo no estructurado. Es importante entender que el servicio de streams de TCP/IP no esta obligado a formar streams estructurados de datos. Por ejemplo, no existe forma para que una aplicación de nómina haga que un servicio de streams marque fronteras entre los registros de empleado o que identifique el contenido del stream como datos de nómina. Los programas de aplicación que utilizan el servicio de stream deben entender el contenido del stream y ponerse de acuerdo sobre su formato antes de iniciar una conexión.

- Conexión Full Duplex. Las conexiones proporcionadas por el servicio de streams de TCP/IP permiten la transferencia concurrente en ambas direcciones. Dichas conexiones se conocen como full duplex. Desde el punto de vista de un proceso de aplicación, una conexión full duplex consiste en dos flujos independientes que se mueven en direcciones opuestas, sin ninguna interacción aparente. El servicio de streams permite que un proceso de aplicación termine el flujo en una dirección mientras los datos continúan moviéndose en la otra dirección, haciendo que la conexión sea half duplex. La ventaja de una conexión full duplex es que el software subyacente de protocolo puede enviar en datagramas información de control de flujo al origen, llevando datos en la dirección opuesta. Este procedimiento de carga, transporte y descarga reduce el trafico en la red.

Hemos dicho que el servicio de entrega de flujo confiable garantiza la entrega de los datos enviados de una maquina a otra sin perdida o duplicación. Surge la pregunta: "¿Cómo puede el software de protocolo proporcionar una transferencia confiable si el sistema subyacente de comunicación ofrece una entrega no confiable de paquetes?" La respuesta es complicada, pero la mayor parte de los protocolos confiables utilizan una técnica fundamental conocida como acuse de recibo positivo con retransmisión. La técnica requiere que un receptor se comunique con el origen y le envíe un mensaje de acuse de recibo (ACK de Acknowledge) conforme recibe los datos. El transmisor guarda un registro de cada paquete que envía y espera un acuse de recibo antes de enviar el siguiente paquete. El transmisor también arranca un temporizador cuando envía un paquete y lo retransmite si dicho temporizador expira antes de que llegue un acuse de recibo.

En la figura 3.10 se muestra cómo transfiere datos el protocolo mas sencillo de acuse de recibo positivo.

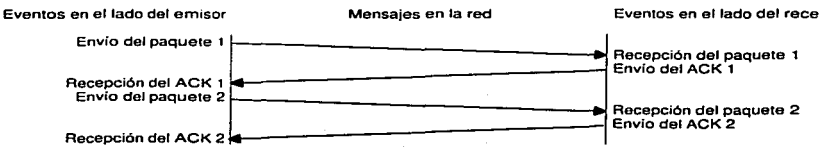


Figura 3.10 Un protocolo que se vale de acuses de recibo positivos con retransmisión.
En este protocolo, el emisor espera un acuse de recibo para cada paquete enviado. La distancia vertical bajo la figura representa el incremento en el tiempo y las líneas que cruzan en diagonal representan la transmisión de paquetes de red.

En la figura, los eventos en el transmisor y receptor se muestran a la izquierda y derecha, respectivamente. Cada línea diagonal que cruza por el centro muestra la transferencia de un mensaje a través de la red.

En la figura 3.11 se utiliza el mismo diagrama de formato que en la figura 3.10 para mostrar que sucede cuando se pierde o corrompe un paquete. El transmisor arranca un temporizador después de enviar el paquete. Cuando termina el tiempo, el transmisor asume que el paquete se perdió y lo vuelve a enviar.

El problema final de confiabilidad surge cuando un sistema subyacente de entrega de paquetes los duplica. Los duplicados también pueden surgir cuando las redes tienen grandes retrasos que provocan la retransmisión prematura. La solución de la duplicación requiere acciones cuidadosas ya que tanto los paquetes como los acuses de recibo se pueden duplicar. Por lo general, los protocolos confiables detectan los paquetes duplicados al asignar a cada uno un numero de secuencia y al obligar al receptor a recordar que números de secuencia recibe. Para evitar la confusión causada por acuses de recibo retrasados o duplicados, los protocolos de acuses de recibo positivos envían los números de secuencia dentro de los acuses, pare que el receptor pueda asociar correctamente los acuses de recibo con los paquetes.

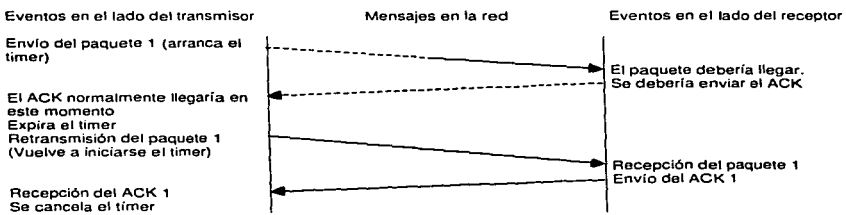


Figura 3.11 Tiempo excedido y retransmisión que ocurre cuando un paquete se pierde.
La línea punteada muestra el tiempo que podría ocuparse para la transmisión de un paquete y su acuse de recibo, si no se perdiera el paquete.

3.4.2. Ventanas deslizables

Antes de examinar el servicio de streams de TCP, necesitamos explorar un concepto adicional que sirve de base pare la transmisión de streams. Este concepto, conocido como ventana deslizable, hace que la transmisión de streams sea eficiente. Para entender lo que motiva a utilizar ventanas deslizables, hay que recordar la secuencia de eventos que se muestran en la figura 3.10.

A fin de lograr la confiabilidad, el transmisor envía un paquete y espera un acuse de recibo antes de enviar otro. Como se muestra en la figura 3.10, los datos sólo fluyen entre las maquinas en una dirección a la vez, inclusive si la red tiene capacidad para comunicación simultanea en ambas direcciones. La red estará del todo ociosa durante el tiempo que las maquinas retrasen sus respuestas (por ejemplo, mientras las máquinas computan rutas o sumas de verificación). Si nos imaginamos una red con altos retrasos en la transmisión, el problema es evidente:

Un protocolo simple de acuses de recibo positivos ocupa una cantidad sustancial de ancho de banda de red debido a que debe retrasar el envío de un nuevo paquete hasta que reciba un acuse de recibo del paquete anterior.

La técnica de ventana deslizable es una forma mas compleja de acuse de recibo positivo y retransmisión que el sencillo método mencionado antes. Los protocolos de ventana deslizable utilizan el ancho de banda de red de mejor forma, ya que permiten que el transmisor envíe varios paquetes sin esperar un acuse de recibo. La manera mas fácil de visualizar la operación de ventana deslizable es pensar en una secuencia de paquetes que se transmitirán como se muestra en la figura 3.12.

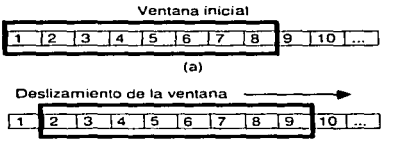


Figura 3.12. (a) Un protocolo de ventana deslizante con ocho paquetes en la ventana y (b) La venta deslizada.

La ventana se desliza hacia el paquete 9, este puede enviarse cuando se recibe un acuse de recibo del paquete 1. Unicamente se retransmiten los paquetes sin acuse de recibo.

El protocolo coloca una ventana pequeña y de tamaño fijo en la secuencia, y transmite todos los paquetes que residan en la ventana. Decimos que un paquete esta marcado como sin acuse de recibo si se transmitió pero no se recibió ningún acuse de recibo. Técnicamente, el número de paquetes sin acuse de recibo en un tiempo determinado depende del tamaño de la ventana y está limitado a un número pequeño y fijo. Por ejemplo, en un protocolo de ventana deslizable de 8, se permite al transmisor enviar 8 paquetes antes de recibir un acuse de recibo.

Como se muestra en la figura 3.12, una vez que el transmisor recibe un acuse de recibo para el primer paquete dentro de la ventana, "mueve" la misma y envía el siguiente paquete. La ventana continuará moviéndose en tanto se reciban acuses de recibo. El desempeño de los protocolos de ventana deslizable depende del tamaño de la ventana y de la velocidad con que la red acepta paquetes.

En la figura 3.13 se muestra un ejemplo de la operación de un protocolo de ventana deslizable cuando se envían tres paquetes. Nótese que el transmisor los envía antes de recibir cualquier acuse de recibo. Con un tamaño de ventana 1, un protocolo de ventana deslizable sería idéntico a un protocolos simple de acuse de recibo positivo. Al aumentar el tamaño de la ventana, es posible eliminar completamente el tiempo ocioso de la red. Esto es, en una situación estable, el transmisor puede enviar paquetes tan rápido como la red los pueda transferir.

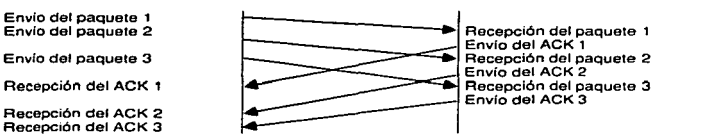


Figura 3.13 Ejemplo de tres paquetes transmitidos mediante un protocolo de ventana deslizante.
El concepto clave es que el emisor pueda transmitir todos los paquetes de la ventana sin esperar un acuse de recibo.

Ya que comprendimos el principio de las ventanas deslizables, podemos examinar el servicio de flujo confiable proporcionado por la suite de protocolos de TCP/IP. TCP es complejo, el protocolo especifica el formato de datos y los acuses de recibo que intercambian dos computadoras para lograr una transferencia confiable, así como los procedimientos que la computadora utiliza para asegurarse de que los datos lleguen de manera correcta. También, especifica cómo el software TCP distingue el correcto entre muchos destinos en una misma máquina, y cómo las máquinas en comunicación resuelven errores como la pérdida o duplicación de paquetes. El protocolo especifica también como dos computadoras inician una transferencia de streams de TCP y como se ponen de acuerdo cuando se completa.

Asimismo, es importante entender lo que el protocolo no incluye. Aunque la especificación TCP describe como utilizan a TCP los programas de aplicación en términos generales, no aclara los detalles de la interfaz entre un programa de aplicación y TCP. Esto es, la documentación del protocolo sólo analiza las operaciones que TCP proporciona; no especifica los procedimientos exactos que los programas de aplicación invocan para tener acceso a estas operaciones. La razón para no especificar la interfaz del programa de aplicación es la flexibilidad. En particular, debido a que los programadores por lo general implantan TCP en el sistema operativo de una computadora, necesitan emplear la interfaz que proporciona el sistema operativo, sea cual sea. Permitir que la implantación tenga flexibilidad hace posible tener una sola especificación para que TCP pueda utilizarse para el diseño de software en una gran variedad de máquinas.

Debido a que TCP asume muy poco sobre el sistema subyacente de comunicación, TCP se puede utilizar con una gran variedad de sistemas de entrega de paquetes, incluyendo el servicio de entrega de paquetes, incluyendo el servicio de entrega de datagramas IP.

Por ejemplo, TCP puede implantarse para utilizar líneas telefónicas, una red de área local, una red de fibra óptica de alta velocidad o una red de largo recorrido y baja velocidad. De hecho, la gran variedad de sistemas de entrega que puede utilizar TCP le ha dado gran éxito.

Al igual que UDP, TCP reside sobre IP en el esquema de estratificación por capas de protocolos. TCP permite que varios programas de aplicación en una máquina se comuniquen de manera concurrente y realiza el demultiplexado del tráfico TCP entrante entre los programas de aplicación. Al igual que UDP, TCP utiliza números de puerto de protocolo para identificar el destino final dentro de una máquina. Cada puerto tiene asignado un número entero pequeño utilizado para identificarlo.

Cuando hablamos de los puertos de UDP, dije que se podría pensar que cada puerto es una cola de salida en la que el software de protocolo coloca los datagramas entrantes. Sin embargo, los puertos de TCP son mas complejos, ya que un número de puerto no corresponde a un solo objeto. De hecho, TCP se diseño según la abstracción de conexión, en la que los objetos que se van a identificar son conexiones de circuito virtual, un puertos individuales. Entender que TCP utiliza la noción de conexiones es de vital importancia, ya que nos ayuda a explicar el significado y la utilización de los números de puerto TCP.

Una conexión consiste de un circuito virtual entre dos programas de aplicación, por esta razón, puede ser natural asumir que un programa de aplicación sirve como el "punto extremo" de la conexión. Sin embargo, no es así. TCO define que un punto extremo es un par de números enteros (host, puerto) en donde el host se identifica por su dirección IP. Por ejemplo, el punto extremo (132.248.63.20,21) se refiere al puerto TCP 21 en la máquina con dirección 132.248.63.20.

Las conexiones se definen por sus puntos finales, por ejemplo, si tenemos una conexión entre dos máquinas de la UNAM, esta se definiría por los siguientes puntos finales:

(132.248.56.124,40) ◄——————► (132.248.72.10,1469)

Hasta el momento sólo se han utilizado direcciones de puerto únicas para cada conexión, sin embargo se puede dar el caso de que dos conexiones utilicen el mismo numero de puerto. En un principio podría parecer ambiguo, sin embargo, no existe tal ambigüedad, ya que TCP se basa en conexiones y no sólo en números de puerto. De otra forma se tendría que generar números de puerto para cada conexión realizada.

A diferencia de UDP, TCP es un protocolo orientado a conexión que requiere que ambos extremos establezcan comunicación. En TCP, los puntos extremos deben de establecer una conexión antes de poder enviar cualquier dato. Para establecer una conexión, el programa de aplicación realiza una función de apertura pasiva al ponerse en contacto con el sistema operativo e indicarle que aceptará una conexión entrante. En ese momento, el sistema operativo asigna un número de puerto TCP a su extremo de la conexión. El programa de aplicación en el otro extremo debe contactar a su sistema operativo mediante una solicitud de apertura activa para establecer una conexión. Los dos módulos de TCP se comunican para establecer y llevar a cabo la comunicación.

TCP visualiza al flujo de datos como una secuencia de bytes que divide en segmentos para realizar su transmisión. Por lo general, cada segmento viaja por la red como un datagrama IP.

TCP utiliza un mecanismo especializado de ventana deslizable para solucionar dos problemas importantes: la transmisión eficiente y el control de flujo. Al igual que el protocolo de ventana deslizable antes descrito, el mecanismo de ventana de TCP hace posible enviar varios segmentos antes de que llegue un ACK. Este mecanismo aumenta la generación total de salida ya que mantiene ocupada la red. El protocolo de ventanas de TCP también soluciona el problema de control de flujo, de extremo a extremo, ya que permite al receptor restringir la transmisión hasta que tenga espacio suficiente en buffer para incorporar mas datos.

El mecanismo TCP de ventana deslizante opera a nivel de octetos, no a nivel de segmento ni de paquete. Los octetos del flujo de datos se numeran de manera secuencial, y el transmisor guarda tres apuntadores asociados a cada conexión. Los apuntadores definen una ventana deslizante como la que se muestra en la figura 3.14. El primer apuntador marca el extremo izquierdo de la ventana, separa los octetos que ya se enviaron y se recibió su acuse de recibo de los que ya se enviaron y no se ha recibido un acuse de recibo. El segundo apuntador marca el extremo derecho de la ventana deslizable y define el octeto más alto en la secuencia que se puede enviar antes de recibir mas acuses de recibo. El tercer apuntador señala la frontera dentro de la ventana que separa los octetos que ya se enviaron de los que todavía no se envían. El software de protocolo envía sin retraso todos los octetos dentro de la ventana, por lo general, la frontera dentro de la ventana se mueve rápidamente de izquierda a derecha.

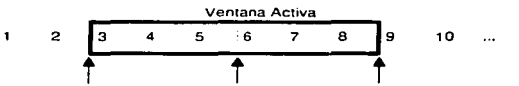


Figura 3.14 Ejemplo de una ventana deslizante de TCP

Los octetos hasta de no. 2 se han enviado y se tiene su ACK, los octetos del 3 al 6 han sido enviados pero no reconocidos, los octetos del 7 al 9 no se han enviado pero serán enviados sin retardo y los octetos del 10 en adelante no pueden ser enviados hasta que la ventana se mueva.

Se debe tener en cuenta que la ventana no sólo se tiene en el transmisor, sino que también se localiza en el receptor. Dado que la comunicación en TCP es full duplex, se debe tomar en cuenta que cada extremo de la conexión debe enviar y recibir información. Por esta razón, cada extremo tiene 2 ventanas de control de flujo.

Una diferencia entre el protocolo TCP de ventana deslizable y el protocolo simplificado de ventana deslizable presentado anteriormente, es que TCP permite que el tamaño de la ventana varíe.

Cada acuse de recibo, informa cuantos octetos se recibieron y además contiene un aviso de ventana que especifica cuantos octetos adicionales de datos esta preparado para aceptar el receptor. Pensemos el aviso de ventana como la especificación del tamaño actual del buffer del receptor. En respuesta a un aumento en el aviso de la ventana, el transmisor aumenta el tamaño de ventana, el transmisor aumenta el tamaño de su ventana deslizable y procede al envío de octetos de los que todavía no se tiene un acuse de recibo. En respuesta a una disminución en el aviso de ventana, el transmisor disminuye el tamaño de su ventana y deja de enviar los octetos que se encuentran mas allá de la frontera.

La ventaja de utilizar una ventana de tamaño variable es que esta proporciona control de flujo así como una transferencia confiable. Si el buffer receptor se llena, no puede aceptar mas paquetes, así que envía un anuncio de ventana mas pequeño. En caso extremo, el receptor anuncia una ventana de tamaño cero para detener la transmisión. Posteriormente cuando se tiene espacio suficiente en el buffer, el receptor anuncia un tamaño de ventana distinto de cero.

Tener un mecanismo de control de flujo es de vital importancia en un ambiente como Internet, ya que las computadoras que se comunican tienen diferentes velocidades y tamaños. Si no existieran mecanismos de control de flujo, al tratar de comunicar una minicomputadora con una maiframe, la minicomputadora podría saturar sus buffers demasiado rápido.TCP emplea el esquema de ventana deslizable para resolver el problema de control de flujo en los puntos finales, sin embargo, no cuenta con un mecanismo explícito para el control de congestionamientos.

3.4.3. Formato del segmento de TCP

La unidad de transferencia entre el software TCP de dos máquina se conoce como segmento. Los segmentos se intercambian para establecer conexiones, transferir datos, enviar acuses de recibo, anunciar los tamaños de ventanas y para cerrar conexiones. Debido a que TCP utiliza acuses de recibo incorporados, un acuse que viaja de la máquina A a la máquina B puede viajar en el mismo segmento en el que viajan los datos de la máquina A a la máquina B puede viajar en el mismo segmento en que viajan los datos de la máquina A a la máquina B, aun cuando el acuse de recibo se refiera a datos enviados de B hacia A. En la figura 3.15 se muestra el formato del segmento TCP.

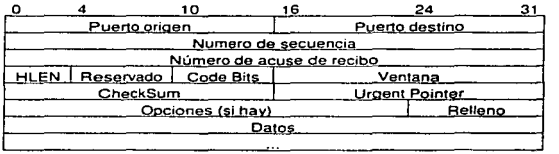


Figura 3.15. Formato de un segmento TCP.

Los segmentos se utilizan para establecer conexiones, así como para transportar datos y acuses de recibo. Obsérvese que el segmento también tiene una zona de encabezado y otra de datos.

El encabezado de TCP transporta la identificación y la información de control. Los campos de puerto origen y puerto destino contienen los números de puerto TCP que identifican a los programas de aplicación en los extremos de la conexión. El campo de número de secuencia identifica la posición de los datos del segmento en el flujo de datos en el transmisor. El campo número de acuse de recibo identifica el número de octetos que la fuente espera recibir después. Obsérvese que el número de secuencia se refiere al flujo que va en la dirección opuesta al segmento.

El campo hlen contiene un número entero que especifica la longitud del encabezado del segmento, medida en múltiplos de 32 bits. Es necesario porque el campo de opciones varía en su longitud, dependiendo de que opciones se seleccionen. El campo de 6 bits marcado como reservado, esta pensado para su uso en un futuro.

Algunos segmentos sólo llevan un acuse de recibo y otros solamente llevan datos. Otros llevan solicitudes para establecer o cerrar una conexión. El software TCP utiliza el campo de 6 bits llamado code bits, para determinar el propósito y contendido del segmento. Los seis bits indican como interpretar otros campos en el encabezado, de acuerdo con la tabla de la figura 3.16.

Bit (de izquierda a derecha)	Significado si el bit tiene valor de 1
URG	El campo de Urgent Pointer es válido
ACK	El campo de acuse de recibo es válido
PSH	Este segmento solicita una operación Push
RST	Inicialización de la conexión
SYN	Sincronizar los números de secuencia
END	El emisor ha llegado al final de su flujo de octetos

Figura 3.16. Significado de los bits del campo CODE BITS en el encabezado de TCP

El software TCP informa sobre cuantos datos está dispuesto a aceptar cada vez que encía un segmento, al especificar su tamaño en buffer en el campo de ventana. El campo contienen un número entero sin signo de 16 bits en orden de octetos estándar de rd. Los anuncios de ventana proporcionan otro ejemplo de acuse de recibo de carga, transporte y descarga ya que acompañan a todos los segmentos, tanto a los que llevan datos, como a los que sólo llevan un acuse de recibo.

3.4.3.1 Datos Urgentes

Aunque TCP es un protocolo orientado al flujo, algunas veces es importante que el programa en un extremo de la conexión envíe datos fuera de banda, sin esperar a que el programa en el otro extremo de la conexión consuma los octetos que están en flujo. Por ejemplo, cuando se utiliza TCP para una sesión de acceso remoto, el usuario puede decidir si envía una secuencia de teclado que interrumpa o aborte el programa en el otro extremo.

Las señales se deben enviar sin esperar a que el programa lea los octetos que ya están en el flujo TCP (o de lo contrario, podría ser imposible interrumpir programas que dejen de leer la entrada).

Para incorporar la señalización fuera de banda, TCP permite que el transmisor especifique los datos como urgentes, dando a entender que se debe notificar su llegada al receptor tan pronto como sea posible, sin importar su posición en el flujo. El protocolo especifica que, cuando se encuentra con datos urgentes, la parte de TCP receptora debe notificar al programa de aplicación que esté asociado con la conexión que inicie la modalidad de "urgente". Después de asimilar todos los datos urgentes, TCP indica al programa de aplicación que regrese a su operación normal.

Los detalles exactos de cómo TCP informa al programa de aplicación sobre datos urgentes dependen del sistema operativo de la máquina. El mecanismo utilizado para marcar los datos urgentes cuando se transmiten en un segmento consiste en el bit de código URG y en el campo urgent pointer. Cuando se activa el bit URD, el urgent pointer especifica la posición dentro del segmento en la que terminan los datos urgentes.

3.4.3.2 Tamaño máximo de segmento

No todos los segmentos que se envían a través de una conexión serán del mismo tamaño. Sin embargo, ambos extremos necesitan acordar el tamaño máximo de los segmentos que se transferirán. El software TCP utiliza el campo de opciones para negociar con el software TCP del otro extremo de la transmisión; una de las opciones permite que el software TCP especifique el tamaño máximo de segmento que está dispuesto a recibir. Por ejemplo, cuando un sistema incorporado que solamente tiene unos cientos de octetos de capacidad en buffers se conecta con una supercomputadora puede negociar un tamaño máximo de segmentos a transferir que restrinja los segmentos para que quepan en el buffer. Para las computadoras conectadas por redes de área local de alta velocidad es especialmente importante escoger un tamaño máximo de segmento que lleno los paquetes o no harán un buen uso del ancho de banda. Por lo tanto, si los dos puntos extremos residen en la misma red física, TCP por lo general computará un tamaño máximo de segmento de tal forma que los datagramas IP resultantes corresponda con el valor del MTU. Si los dos puntos no residen en la misma red física, pueden intentar descubrir la MTU mínima a lo largo del camino entre ellos o pueden escoger un tamaño máximo de segmento de 536 (tamaño máximo asignado por default a un datagrama IP, 576 menos el tamaño estándar de los encabezados IP y TCP).

En un ambiente de Internet, escoger un tamaño máximo de segmento apropiado puede ser difícil, ya que el desempeño puede ser bajo tanto por tamaños de segmento demasiado grandes, como por tamaños muy pequeños. Por una parte, cuando el tamaño del segmento es pequeño, la utilización de la red permanece baja. Para entender por qué, recuerde que los segmentos TCP viajan encapsulados dentro de datagramas IP, que a su vez están encapsulados en frames. Por lo tanto, cada segmento tiene al menos 40 octetos de encabezados TCP e IP, además de los datos. Los datagramas que sólo llevan un octeto de datos utiliza como máximo 1/41 del ancho de banda de la red subyacente para los datos de usuario; en la práctica, las brechas mínimas entre paquetes y el hardware de rd que ponen bits en frames hacen que el rango sea aún mas pequeño.

Por otro lado, los tamaños de segmento muy grandes también pueden producir un bajo desempeño. Los grandes segmentos resultan en grandes datagramas IP. Cuando dichos datagramas viajan a través de la red con un MTU muy pequeño, IP debe fragmentarlos. A diferencia de un segmento TCP, un fragmento no se puede confirmar o retransmitir en forma independiente; todos los fragmentos deben llegar o de lo contrario, se tendrá que transmitir todo el datagrama. Debido a que la probabilidad de perder un datagrama no es cero, aumentar el tamaño del segmento por arriba del umbral de fragmentación disminuye la probabilidad de que lleguen los datagramas, lo cual disminuye la capacidad de salida. En teoría, el tamaño óptimo de segmento se logra cuando los datagramas IP son tan grandes como sea posible sin requerir fragmentación.

3.4.3.3 Cálculo del Checksum de TCP

El campo de checksum del encabezado de TCP contiene la suma de verificación de números enteros y 16 bits que se utiliza para verificar la integridad de los datos así como del encabezado de TCP. Para realizar el cálculo del checksum, TCP realiza un procedimiento similar al de UDP de agregar un pseudo-encabezado. Al igual que UDP, los campos que se agregan tampoco se transmiten.

El propósito de utilizar el pseudo-encabezado es exactamente el mismo que en UDP. Permite al receptor verificar la integridad del segmento. Al igual que en UDP, al generar el pseudo-encabezado de TCP, se requiere hacer pasar la dirección IP origen y destino. El formato del pseudo-encabezado se presenta en la siguiente figura:

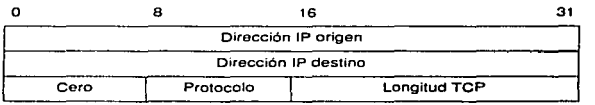


Figura 3.17 Formato del pseudo-encabezado de TCP.

Al momento de construirse el pseudo-encabezado, el transmisor asigna al campo de protocolo el valor que utilizará la capa adyacente en el campo de tipo de protocolo. Para los datagramas IP que transportan TCP, el valor es de 6. El campo longitud TCP especifica la longitud total del segmento TCP, incluyendo el encabezado TCP. En el extremo receptor, la información utilizada en el pseudo-encabezado se extrae del datagrama IP que transportó el segmento y se incluye en el cálculo del checksum.

3.4.4. Retransmisión de paquetes

Como TCP envía los datos en segmentos de longitud variable, y debido a que los segmentos retransmitidos pueden incluir mas datos que los originales, los acuses de recibo no pueden remitirse fácilmente a los datagramas o segmentos. De hecho, se remiten a una

posición en el flujo, utilizando los números de secuencia de flujo. El receptor recolecta octetos de datos de los segmentos entrantes y reconstruye una copia exacta del flujo que se envía. Como los segmentos viajan en datagramas IP, se pueden perder o llegar en desorden; el receptor utiliza los números de secuencia para reordenar los segmentos. En cualquier momento, el receptor tendrá cero o mas octetos reconstruidos contiguamente desde el comienzo del flujo, pero puede tener piezas adicionales del flujo de datagramas que hayan llegado en desorden. El receptor siempre acusa de recibido del prefijo contiguo mas largo del flujo que se recibió correctamente. Cada acuse de recibo especifica un calor de secuencia mayor en la unidad, con respecto al octeto de la posición más alta en el prefijo contiguo que recibió. Por lo tanto, el transmisor recibe una retroalimentación continua del receptor conforme progresa el flujo.

Al esquema de acuse de recibo de TCP se le llama acumulativo porque reporta cuanto se ha acumulado del flujo. Los acuses de recibo tiene ventajas y desventajas. Una ventaja es que los acuses de recibo son fáciles de generar y no son ambiguos. Otra es que los acuses de recibo perdidos no necesariamente forzarán la retransmisión. Una gran desventaja es que el receptor no tiene información sobre todas las transmisiones exitosas, sino únicamente sobre la posición en el flujo que se recibió.

Para entender por que la falta de información sobre todas las transmisiones exitosas hace que los acuses de recibo acumulativos sean menos eficientes, piense en una ventana que abarca 5000 octetos comenzando en la posición 101 en el flujo, y suponga que el transmisor envió todos los datos en la ventana al transmitir cinco segmentos. Suponga también que se pierde el primer segmento y todos los demás llegan intactos. Conforme llega cada segmento, el receptor envía un acuse de recibo, pero todos los acuses especifican el octeto 101, que es el octeto contiguo siguiente mas alto que espera recibir. No hay forma para que el receptor indique al transmisor que llegó la mayor parte de los datos para la ventana actual.

Cuando ocurre el fin del tiempo en el extremo transmisor, éste debe escoger entre dos esquemas potencialmente ineficaces. Puede transmitir un segmento o retransmitir los cinco. En este caso, retransmitir los cinco segmentos no es eficaz. Cuando llega el primer segmento, el receptor tendrá todos los datos en la ventana y el acuse de recibo aparecerá 5101. Si el transmisor sigue el estándar aceptado y retransmite sólo el primer segmento para el que no hay acuse, debe esperar a obtener el acuse de recibo antes de decidir qué y cuantos enviar. Por lo tanto regresa a un protocolo simple de acuse de recibo positivo y puede perder las ventajas de una gran ventana.

3.4.5. Establecimiento de una conexión TCP.

Para establecer una conexión, TCP utiliza un hadshake de tres etapas. En el caso mas sencillo, este intercambio procede como se muestra en la figura 3.18.

El primer segmento del handshake se puede identificar porque tiene activo el bit SYN en el campo de code bits. El segundo mensaje tiene tanto el bit SYM como el bit ACK activos, indicando tanto el acuse de recibo del primer segmento como el hecho de que se continúa con el intercambio. El mensaje final del hanshake es sólo un acuse de recibo y

nada mas se utiliza para informar al destino que ambos extremos están de acuerdo en establecer una conexión.

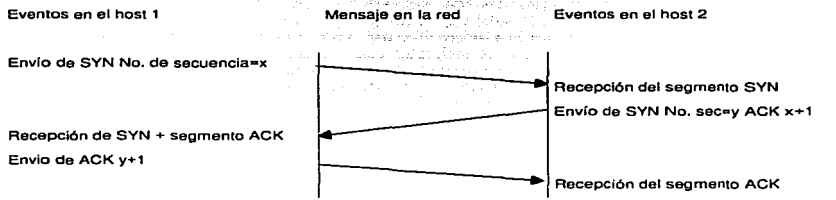


Figura 3.18. Secuencia de mensajes del handshake de tres etapas.
En la representación, el tiempo transcurre hacia la parte inferior; las líneas diagonales representan segmentos enviados entre los hosts. Los segmentos SYN transportan información sobre el número de secuencia inicial.

Por lo general, software TCP en una máquina espera de forma pasiva el intercambio de señales y el software TCP en otra máquina lo inicia. Sin embargo, el handshake esta cuidadosamente diseñado para funcionar aún cuando ambas máquinas intenten iniciar una conexión al mismo tiempo. Por lo tanto, se puede establecer una conexión desde cualquier extremo o desde ambos al mismo tiempo. Una vez que se establece la conexión, los datos pueden fluir en ambas direcciones por igual. No existe un maestro ni un esclavo.

El handshake de tres etapas es necesario y suficiente para la sincronización correcta entre los dos extremos de la conexión. Para entender por qué, se debe recordar que TCP esta sobre un servicio de entrega no confiable de paquetes, así que los mensajes pueden perderse, retrasarse, duplicarse o entregarse fuera de orden. Por lo tanto, el protocolo debe utilizar un mecanismo de terminación de tiempo y retransmitir las solicitudes perdidas. Sucederán algunos problemas si las solicitudes originales y retransmitidas llegan mientras se establece la conexión o si las solicitudes retransmitidas se retrasan hasta que se establezca, utilice y termine la conexión. Un handshake de tres etapas (mas la regla de que TCP ignore las solicitudes adicionales de conexión después de que se establezca la misma) resuelve estos problemas.

El hanshake de tres etapas realiza dos fucniones importantes. Garantiza que ambos lados estén listos para transferir datos (y que tengan conocimiento de que ambos están listos) y permite, a ambas partes, acordar un número de secuencia inicial. Los números de secuencia son enviados y reconocidos durante el handshake. Cada máquina debe seleccionar un número de secuencia inicial en forma aleatoria que se utilizará para identificar octetos en el flujo que se está enviando. Los números de secuencia no pueden empezar siempre con el mismo valor. En particular, TCP no puede seleccionar una secuencia 1 cada vez que crea una conexión. Por supuesto, es importante que ambas partes acuerden un número inicial, así como el número de octetos empleados en un acuse de recibo de acuerdo a los utilizados en el segmento de datos.

3.4.6. Fin de una conexión TCP

Dos programas que utilizan TCP para comunicarse pueden terminar la conversación cortésmente valiéndose de la operación close. De manera más interna, TCP utiliza una modificación del handshake de tres etapas para cerrar las conexiones. Recordemos que las conexiones TCP son de tipo full duplex y que hemos visto que estas contienen dos transferencias de flujo independientes, una en cada dirección. Cuando un programa de aplicación informa a TCP que ya no tiene mas datos para enviar, este cerrara la conexión en una dirección. Para cerrar la mitad de una conexión, el emisor TCP termina de transmitir los datos restantes, espera la recepción de un acuse de recibo y, entonces, envía un segmento con el bit END activado. El receptor TCP reconoce el segmento END e informa al programa de aplicación en su extremo que no tiene mas datos disponibles (por ejemplo, mediante el mecanismo de fin de archivo de sistema operativo).

Una vez que la conexión se ha cerrado en una dirección dada, TCP rechaza mas datos en esta dirección. Mientras tanto, los datos pueden continuar fluyendo en la dirección opuesta hasta que el emisor se cierra. Por su puesto, los acuses de recibo continúan fluyendo hacia el emisor incluso después de que la conexión se ha cerrado. Cuando ambas direcciones se han cerrado, el software TCP en cada punto extremo borra sus registros de la conexión.

Los detalles del cierre de una conexión son más sutiles de lo que se ha sugerido anteriormente porque TCP utiliza un handshake de tres etapas modificado para cerrar una conexión. La figura 3.19 ilustra el procedimiento.

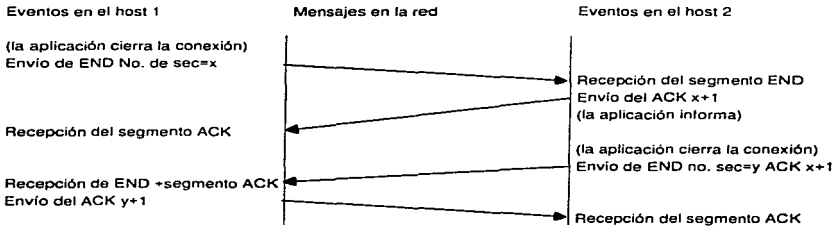


Figura 3.19. Modificación del handshake de tres etapas para cerrar conexiones.
El host que recibe el primer segmento END lo reconoce de inmediato y, después, lo retarda antes de enviar el segundo segmento END.

La diferencia entre el hanshake de tres etapas empleado para establecer e interrumpir conexiones se presenta luego de que la máquina recibe el segmento END inicial. En lugar de genera un segundo segmento END inmediatamente, TCP envía un acuse de recibo y luego informa a la aplicación de la solicitud de interrupción. Informar al programa de aplicación de la solicitud y obtener una respuesta, puede tomar un tiempo considerable (por ejemplo, si comprende la interacción humana).

El acuse de recibo evita la retransmisión del segmento inicial END durante la espera. Por último, cuando el programa de aplicación le envía la orden a TCP que interrumpa la conexión completamente, TCP envía el segundo segmento END y la localidad original responde con el ACK.

3.4.7. Maquina de estados de TCP

Como en la mayor parte de los protocolos, la operación de TCP se puede explicar mejor mediante una maquina de estados finitos. La figura 3.20 muestra la máquina de estados finitos de TCP, en ella, los círculos representan estados y las flechas representa transiciones entre estos.

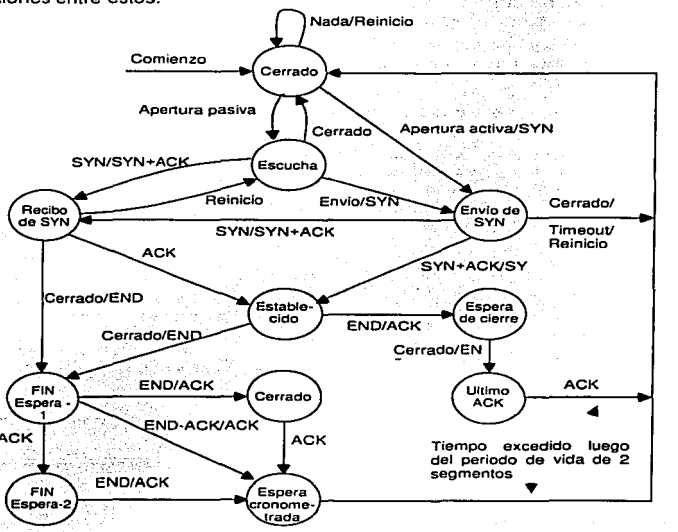


Figura 3.20. Máquina de estados finitos.

Cada punto final comienza en el estado cerrado. Los nombres de las transiciones muestran la entrada que ocasiona la transición seguida por la salida, si la hay.

El nombre de cada transición muestra que recibe TCP para generar la transición y que envía como respuesta. Por ejemplo, el software TCP en cada extremo comienza en un estado cerrado. El programa de aplicación debe emitir un comando de apertura pasiva para iniciar la conexión. El comando de apertura de conexión activa obliga a que se de una transición del estado cerrado al estado enviar SYN. Cuando TCP continua con la transición, emite un segmento SYN. Cuando el otro extremo devuelve que contiene un SYN, mas un ACK, TCP cambia al estado establecido y comienza la transferencia de datos.

Una vez que hemos visto de manera sencilla en funcionamiento de TCP, cabe aclarar que al igual que UDP, TCP también tiene un conjunto de números de puerto reservados. TCP a su vez, combina la asignación dinámica y estática de puertos mediante un conjunto de puertos bien conocidos para programas llamados con frecuencia, pero la salida de la mayor parte de los puertos disponibles para el sistema operativo se asigna conforme los programas lo necesitan. Aún cuando el estándar original reservaba los números de puerto menores a 256 para asignarlos como puertos bien conocidos, ahora se han asignado números superiores a 1024. La figura 3.21 lista algunos de los puertos TCP asignados en la actualidad. Habría que puntualizar que, aunque los números de puerto TCP y UDP son independientes, los diseñadores han decidido utilizar el mismo número de puerto para cualquier servicio accesible desde UCP y TCP.

Decimal	Clave	UNIX	Descripción
0		-	Reservado
1	TCPMUX	-	Multiplexor TCP
5	RJE	-	Petición de función remota
7	ECHO	echo	Eco
9	DISCARD	discard	Descartar
11	USERS	systat	Usuarios Activos
13	DAYTIME	daytime	Fecha y hora
15	-	netstat	Programa de monitoreo de red
17	QUOTE	qotd	Citas
19	CHARGEN	chargen	Generador de caracteres
20	FTP_DATA	ftp-data	FTP de datos
21	FTP	ftp	FTP de comandos
23	TELNET	telnet	Sesión remota
25	SMTP	smtp	Protocolo de transporte de correo sencillo
37	TIME	time	Hora
42	NAMESERVER	name	Servidor de Nombres de máquinas
43	NICNAME	whois	Servidor de Nombres de usuarios
53	DOMAIN	nameserver	Servidor de dominios
79	FINGER	finger	Finger

Figura 3.21 Ejemplos de números de puertos TCP asignados actualmente.

3.5.Dependencias entre protocolos

TCP/IP se ha convertido en el protocolo de comunicación de red mas importante. Gracias a esto, ha sido posible el desarrollo de múltiples aplicaciones basadas en este protocolo. Todas estas aplicaciones han desarrollado su propio protocolo de aplicación y confían en TCP o UDP para el paso de la información de una entidad a otra.

Aunque no es importante comprender los detalles de todos los protocolos, si lo es saber que protocolos existen y cómo se pueden usar. Aquí se proporciona un breve resumen de las relaciones entre los principales protocolos y se muestra cuáles están disponibles para usarse en programas de aplicación.

En la figura 3.22 se muestra las dependencias entre los principales protocolos. Cada recuadro corresponde a un protocolo y está colocado directamente encima de los protocolos que utiliza.

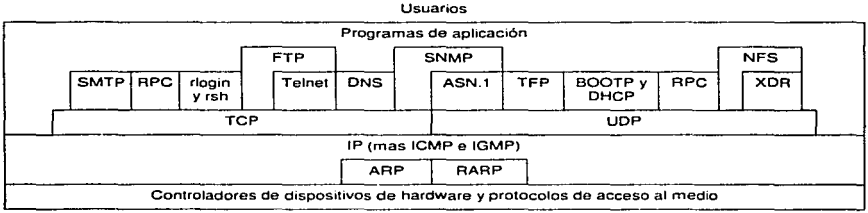


Figura 3.22 Dependencias entre los principales protocolos de TCP/IP de más alto nivel.
Un protocolo utiliza los protocolos que dependen directamente de él. Los programas de aplicación pueden utilizar todos los que estén por encima de IP.

Para varias partes del diagrama se necesita una mayor explicación. La capa inferior representa todos los protocolos que proporciona el hardware. Este nivel comprende cada uno de los protocolos de control de hardware, así como los rangos de acceso a medios hacia la ubicación de enlace lógico. En los capítulos anteriores he asumido que cualquier sistema de transferencia de paquetes puede incluirse en esta capa en tanto que IP pueda utilizarlo para transferir datagramas. De ese modo, si un sistema se configura para mandar datagramas a través de un túnel, la entrada al túnel se considera como una interfaz de hardware, sin importar su implantación de software.

La segunda capa está integrada por las listas inferiores de ARP y RARP. Por supuesto, no todas las máquinas o tecnologías de red lo utilizan. ARP es el más utilizado en Ethernet; RARP se emplea en raras ocasiones salvo en el caso de las máquinas sin disco. Puede haber algunos otros protocolos de enlace de direcciones, pero ninguno tiene un uso amplio.

La tercera capa de la parte inferior contiene a IP. Comprende el protocolo de mensajes de error y control requerido (ICMP) y el protocolo de administración de grupos opcionales de multidifusión (IGMP). Hay que observar que IP es el único protocolo que ocupa toda la capa. Los protocolos de más bajo nivel entregan información que llega de IP y los de más alto nivel deben utilizar IP para enviar datagramas. IP se muestra con una dependencia directa de la capa de hardware, ya que necesita utilizar el enlace de hardware o los protocolos de acceso para transmitir datagramas después de utilizar ARP para direcciones enlazadas.

TCP y UDP componen la capa de transporte, y aunque se han propuesto nuevos estándares para esta capa, ninguno ha sido aceptado por completo.

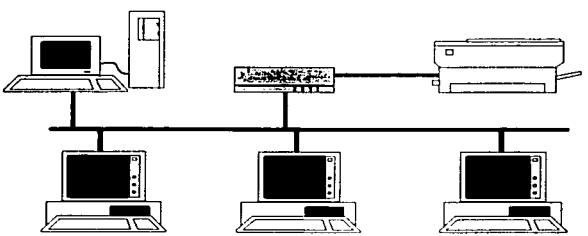
La capa de aplicación ilustra las complejas dependencias entre los diversos protocolos de aplicación. FTP emplea las definiciones de la terminal virtual de red de Telnet para definir la comunicación en su conexión de control y a TCP para formar conexiones de datos. Así pues, el diagrama muestra que FTP depende de telnet y de TCP. El sistema de nombres de dominios (DNS) se vale de UDP y TCP para la comunicación, de modo que el diagrama muestra ambas dependencias.

La mayor parte de los sistemas restringe los programas de aplicación de acceso a los protocolos de nivel inferior. Por lo general, un programa de aplicación puede emplear TCP o UDP, o bien implantar protocolos de mas alto nivel que los utilicen (como FTO). Una aplicación puede necesitar un privilegio especial para abrir puertos específicos, pero esto es completamente diferente del acceso restringido. Algunos sistemas no tienen mecanismos que permitan a un programa de aplicación acceder a IP de manera directa; casi ninguno permite que los programas de aplicación accedan a protocolos como ARP. A pesar de las limitaciones usuales, el diagrama sugiere que las aplicaciones pueden acceder a IP.

- TCP/IP para DOS
- Windows en Red

4

Implantación de TCP/IP en equipos personales



Lista de Figuras

Ya hemos visto el funcionamiento de los protocolos de la suite TCP/IP, sin embargo, TCP/IP fue diseñado y desarrollado primeramente en ambientes UNIX. Anteriormente, UNIX era el sistema operativo por excelencia y la mayoría de las computadoras en las que era implantado eran mainframes ó equipos de alto costo. Con el surgimiento de las computadoras personales, se dio toda una revolución en las computadoras y en la forma misma de concebir a las computadoras.

Las PC's llegaron a todos los rincones del mundo, para estas se crearon sistemas operativos, programas y un sin fin de accesorios y aplicaciones diversas. Con el surgimiento de las redes, los investigadores idearon la forma de comunicar máquinas similares por medio de enlaces punto a punto. La red creció y pronto habia cientos de computadoras conectadas entre sí. Las PC's se comenzaron a unir a las redes, y para ellas se comenzaron a crear distintos programas para poder tener acceso a los recursos que la red presentaba.

En este capítulo se presentarán los mecanismos más importantes para la utilización de computadoras personales en ambientes de red TCP/IP.

4.1. TCP/IP para DOS

Dos es probablemente el sistema operativo mas ampliamente usado en el mundo entero. Aunque sus creadores, IBM y Microsoft han tratado de sustituirlo poniendo en el mercado OS/2 y Windows NT, DOS sigue estando a la cabeza, aunque con la llegada de Windows 95, la gran mayoría de las aplicaciones se están orientando al nuevo sistema operativo.

DOS fue diseñado para el procesador 8088, es pequeño, rápido y hace buen uso de los recursos. El buen uso de los recursos fue de vital importancia, debido a que en los principios de las PC's, estos eran realmente limitados. No habían ciclos de CPU consumidos por sobrecarga innecesaria del sistema operativo, así que cualquier aplicación corriendo en DOS tenía todo el sistema a su disposición. Sin embargo, DOS no es la panacea. DOS también tiene sus limitaciones y fallas.

DOS no soporta multitarea o gran cantidad de memoria. En un principio, esto no era tanto problema, pero en cuanto surgieron PC's mas poderosas, DOS comenzó a verse pequeño. IBM y Microsoft se unieron para desarrollar un sistema operativo que permitiera realizar multitareas y soportar mayores cantidades de memoria. El resultado del desarrollo fue OS/2.

OS/2 no tuvo éxito. Esto se debió principalmente a la falta de aplicaciones nativas para este sistema operativo, dado que era de nueva creación, las aplicaciones necesitaban ser creadas desde cero. Aunque este sistema operativo permitía la ejecución de programas DOS, estos corrían mas lento que en DOS. Esto tiene cierta lógica, dado que DOS es un sistema operativo ligero y sin mayores complicaciones, mientras que los sistemas operativos que surgieron después integraron nuevas características que van haciendo al software mas pesado para el procesador.

Otro factor que influyó en el éxito de DOS fue el bajo costo que implica. DOS corre en sistemas PC bastante pequeños, mientras que los sistemas operativos multitarea requieren

procesadores muy poderosos y gran cantidad de memoria, lo cual dispara los precios en forma dramática.

Hay muchas implantaciones de TCP/IP para DOS. Hay productos de Novell, FTP software, Sunsoft, Beame & Whiteside, Wollongong y DEC.

4.1.1. Parámetros comunes de la configuración de TCP/IP en DOS

DOS fue diseñado para correr una tarea a un mismo tiempo, pero en un ambiente de red, no se puede esperar hasta que la tarea actual se complete. Hay ocasiones en que la red necesita servicio inmediato: datos que llegan de la red o de la aplicación de usuario y debe ser procesada al instante. El software de red debe estar instalado y listo para procesar los datos cuando sea necesario. La mayoría de los paquetes de TCP/IP para DOS cubren esta necesidad por medio de dispositivos controladores manejados por interrupciones y programas TSR.

La figura 4.1 muestra la posición que guarda TCP/IP dentro de la estructura de PC/NFS.

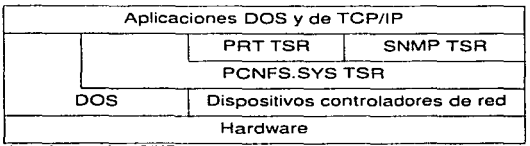


Figura 4.1 Módulos de software de PC/NFS

La figura nos muestra como se ve TCP/IP instalado en un sistema DOS. Muestra los módulos que son específicos a PC/NFS de Sunsoft. Sin embargo, la figura ilustra la estructura de los dispositivos controladores y TSRs comunes a todas las implantaciones deTCP/IP. Los nombres y funciones de cada módulo varían de implantación en implantación, pero los componentes usados para implantar TCP/IP se mantienen igual. Los tres componentes de los que hablo son interrupciones, dispositivos controladores y programas TSR.

4.1.1.1 Interrupciones

Las PC son controladas por interrupciones. Una interrupción es una señal al CPU que le pide suspender la tarea actual y cambiar a otra tarea. Las interrupciones le permiten al hardware y software hacer peticiones inmediatas de servicio al CPU. Impresión, lectura y escritura de archivos y cientos de funciones son posibles gracias a las interrupciones, sin ellas, DOS no podría funcionar.

Las tarjetas de red utilizan interrupciones de hardware para indicar al sistema cuando requieren de un servicio. Las IRQ son sólo el principio, ya que hay mas interrupciones que IRQ , hay 256 interrupciones contra 15 IRQ en la mayoría de los sistemas.

Las interrupciones de software son llamadas con el comando de máquina INT. Las interrupciones de hardware son señalizadas eléctricamente hacia el CPU por medio de pines de salida. Los pines utilizados son:

NMI (Non-Maskable Interrupt)

NMI es la interrupción reservada para problemas catastróficos, tales como sobre carga de voltaje. Como su nombre lo indica, esta interrupción no puede ser enmascarada. Las interrupciones son enmascaradas para permitir que las piezas críticas de código se completen. Las interrupciones no enmascaradas toman precedencia sobre el código crítico y es atendido sin importar que la mascara este habilitada.

INTR

INTR es la interrupción estándar de hardware. Esto es, la línea de interrupción utilizada por el controlador periférico de interfaces (PIC) 8259 para indicar al CPU que se requiere procesar una petición de IRQ

El CPU procesa las interrupciones tan rápido como le es posible y la pasa el control al manejador específico de la interrupción. El CPU guarda el contexto del sistema y entonces salta al manejador de interrupciones apropiado usando el vector almacenado en la tabla de interrupciones. Se pasa un valor de 8 bits al PIC por medio del bus (en el caso de una interrupción de hardware) o pasa como argumento del comando INT (en el caso de una interrupción por software) para determinar cual vector es utilizado. El valor es usado como un índice dentro de la tabla de interrupciones para obtener la dirección correcta del vector.

Hay tres valores comúnmente encontrados en una tabla de interrupciones:

Vector de Interrupción. Una entrada en la tabla que es usada como un vector de interrupción contiene la dirección de un punto de entrada a una rutina manejadora de interrupciones en la forma de un par registro de segmento/registro offset (S:O).

Vectores reservados. Algunos vectores son reservados para uso futuro.

Vector Nulo. Es un vector que contiene el valor de 0000:0000.

4.1.1.2 Programas TSR

Los programas TSR son programas que permanecen en memoria después de regresar el control a DOS. Los programas TSR de TCP/IP son cargados normalmente en el sistema al momento del arranque en el AUTOEXEC:BAT. Los TSR ejecutan una pequeña rutina de inicialización que configura sus vectores de interrupción, reserva la memoria que necesita y regresa el control a DOS usando la función especial 31h de la interrupción estándar 21h. Esta función especial existe porque los TSR son parte del sistema DOS diseñado para proporcionar una limitada forma de procesamiento en fondo. El clásico ejemplo de un TSR es el comando print de DOS, el cual permite la impresión en fondo mientras DOS continua procesando otros comandos de usuario.

Una vez que el TSR esta residente en memoria, este es llamado via interrupciones. También comparte interrupciones multiplexadas con otros TSRs, inicializa su propio vector en la tabla de interrupciones, o intercepta interrupciones que podrían ser normalmente procesadas por otros TSRs o DOS en sí. Los TSR de red que interceptan peticiones a los servicios de I/O de DOS son llamados redirectores. El módulo PCNFS.SYS en la figura 4.1 trabaja exactamente de esta manera. Intercepta las peticiones de servicio de archivos y las examina para determinar si las peticione deben de ser atendidas sobre la red utilizando NFS.

La gran ventaja de implantar TCP/IP como TSR es la velocidad. El software siempre esta en memoria y esta disponible para atender los servicios en t1iempo real. La desventaja es que el software de red utiliza memoria DOS valiosa. Por esta razón es muy importante utilizar un administrador de memoria.

4.1.1.3 Controladores de dispositivos

TCP/IP corre sobre una amplia variedad de redes debido a que está diseñado para ser independiente de cualquier red física. Aunque no requiere una capa inferior de red especifica, necesita alguna red física para mover los bits de un lugar a otro. Para correr TCP/IP en un sistema DOS debemos instalar un dispositivo controlador para la interfaz de red. Los dispositivos físicos, tales como tarjetas de red, se comunican con DOS y con los programas de aplicación por medio de dispositivos controladores instalables. El hardware físico de red y su dispositivo controlador no es realmente parte del protocolo TCP/IP pero es una parte indispensable de cualquier instalación TCP/IP.

Los dispositivos controladores son una característica poderosa de DOS que hace muy fácil agregar nuevos dispositivos de hardware sin modificar el kernel del sistema operativo. Simplemente se agrega la instrucción DEVICE con las opciones correctas dentro del config.sys y es todo. Por su puesto, el controlador tiene que seguir ciertas reglas para comunicarse apropiadamente con DOS y debe saber los detalles del hardware para poder controlarlo. Las reglas de DOS están claramente definidas y el trabajo de escribir el dispositivo controlador es manejado por el vendedor del hardware.

DOS define como los controladores interactúan con el sistema operativo y como los protocolos de las capas superiores se comunican con los controladores. Cada fabricante es libre de definir su propio controlador dentro de estos lineamientos. En el mundo de la PC donde hay diferentes protocolos de red y cientos de fabricantes de tarjetas, por lo que podría esperarse una gran cantidad de incompatibilidades entre los distintos controladores de dispositivos. Por esta razón dos de los desarrolladores de software de red mas importantes, Novell y Microsoft definieron interfaces estándar para adaptadores de red. Los fabricantes de adaptadores que agregan estos estándares a sus productos, pueden asegurar que su hardware puede trabajar con el software de red de esas compañías.

El estándar definido por Microsoft es NDIS (Network Device Interface Specification) y el estándar de Novell es ODI (Open Datalink Interface). Estas especificaciones son diferentes e incompatibles. La mayoría de las implantaciones de TCP/IP soportan controladores NDIS y ODI y la mayoria de las tarjetas de red tienen ambos tipos de controladores. Ambas interfaces permiten tener múltiples pilas de protocolos. Esto permite que TCP/IP pueda compartir una sola interfaz de red con otros protocolos, tales como Netware.

Aplicaciones TCP/IP	Servicios Netware
Protocolos TCP/IP	Shell de estación de trabajo
Convertidor ODI	Protocolos IPX
Capa de Enlace	
Controlador interfaz de múltiples enlaces	
Tarjeta de Interfaz de Red	

Figura 4.2 Pila múltiple de protocolos.

La figura 4.2 muestra la pila de protocolos compartiendo una sola interfaz de red con los protocolos IPX usando un controlador de dispositivo ODI. La capacidad de correr mas de un stack de protocolos sobre una sola interfaz de red es una propiedad importante. Frecuentemente TCP/IP necesita coexistir con Netware o algún otro protocolo de red.

Para poder decidir cual de los estándares se deben utilizar en una instalación, primero se debe de tomar en cuenta el software que se ocupará en las capas superiores. Sin embargo, hay opciones para poder tener ambas opciones juntas, ya sea teniendo como base a NDIS y agregando un dispositivo intermedio para poder manejar ODI o bien a la inversa.

No todos los dispositivos controladores son instalados usando la sentencia DEVICE en el config.sys. Hay dispositivos controladores residentes para dispositivos tales como consola e impresora que son una parte permanente de DOS. Algunas redes tienen dispositivos controladores en programas TSR iniciados en el autoexec.bat. por lo que no son necesarias las modificaciones al config.sys. Los controladores ODI son instalados por medio de TSR, el efecto es el mismo: un controlador para la interfaz de hardware es instalado en el momento de arrancar y permanece residente en memoria.

4.1.2. Instalación de TCP/IP para DOS

Instalar DOS en un sistema DOS se puede reducir a dos pasos básicos: copiar el software dentro del disco duro y configurar el software para un sistema en específico. Las dos tareas son combinadas algunas veces por un programa especial de instalación.

En algunas ocasiones, el programa de instalación realiza los cambios necesarios en el autoexec.bat y en el config.sys, sine embargo, hay ocasiones en las que se tienen que hacer manualmente.

Debido a que la configuración de los parámetros para cada uno de los dispositivos es muy diversa, se deben de tener en cuenta los parámetros específicos de cada uno de los dispositivos.

Configurar TCP/IP en DOS es todo un reto. Al contrario de UNIX, que usa los mismos comandos de configuración de TCP/IP tanto para BSD como System-V, no hay consistencia en los comandos de configuración entre implantaciones de DOS. Sólo mostraré algunos ejemplos de configuración de algunas implantaciones diferentes, pero la única forma de obtener información completa es obtener la documentación del fabricante.

4.1.2.1 Configuración básica

Ya he dicho que la mayoría de los programas de instalación realizan los cambios en forma automática en el autoexec.bat y en el config.sys. Sin embargo cada vez implantación de TCP/IP tiene su propio archivo de configuración y su propia sintaxis de comandos de configuración. La información requerida es la misma para cada implantación, pero los comandos usados para comunicar la información varía de sistema a sistema.

Para PC/TCP de FTP software, el archivo de configuración es el PCTCP.INI, el cual se muestra a continuación:

```
[pctcp kernel]
serial-number=1111-2222-3333
authentication-key=4444-5555-6666
interface=wd8003 0
[pctcp wd8003 0]
ip-address=132.248.60.158
broadcast-address=132.248.60.255
router=132.248.60.254
subnet-mask=255.255.255.0
irq=3
ram-addr=0xD000
io-addr=0x300
```

El archivo PCTCP.INI puede ser creado manualmente o se puede hacer una copia del archivo template.ini, el cual vienen con el software PC/TCP. El ejemplo anterior muestra claramente donde están los datos estándar a todo el software de TCP/IP. El campo de ip-address contiene la dirección IP de la máquina. El campo de broadcast-address es la dirección de broadcast. El campo de subnet-mask especifica la mascara de red. El router contiene la dirección del gateway por default. Como se ve es fácil modificar los valores dentro de este archivo, sin embargo, hay otros valores que son propios de PCTCP.

La línea `[pctcp kernel]` indica que a continuación están los valores de configuración del kernel. Los paréntesis cuadrados indican encabezados de sección.

Dentro de la sección del kernel, aparece una línea con `serial-number`, el cual indica el número de serie del software y a continuación esta la línea de `authentication key` que es utilizada para validar la licencia de software. El número serial y la llave de autenticación son proporcionadas por el fabricante en la documentación y floppies.

Después de los valores de la licencia esta la línea `interface`. Esta línea indica la interfaz de red que se utilizará. El primer campo identifica el tipo de interfaz de hardware, en este caso es una tarjeta Western Digital WD8003- El segundo campo identifica el número de interfaz. Las interfaces están numeradas del 0 al 5.

`[pctcp wd8003 0]` indica que comienza la sección de configuración de la tarjeta 0, en este caso es la wd8003. Si el nombre de la interfaz y el numero son iguales al que aparece en la línea de interface, esta interfaz de red será la que utilice el kernel.

La línea `irq` indica el numero de IRQ que utilizará el adaptador de red. Este no es un valor de configuración para TCP/IP, sin embargo, si este valor no esta bien configurado, todo el acceso a red se verá afectado. La línea ram-addr se utiliza para configurar la memoria utilizada por el adaptador. La línea io-addr es el puerto de I/O utilizado por la tarjeta.

Un segundo ejemplo de configuración de TCP/IP en DOS es Lan Work Place de Novell, el archivo de configuración NET.CFG se presenta a continuación:

```
Link support
        Buffers      8      1500
        Mempool      4096
 Link Driver NE2000
        Int 5
        Port 320
        Frame Ethernet_II
 Protocol TCPIP
        PATH TCP_CFG        c:\TCP
        Ip_address  132.248.60.158
        Ip_router   132.248.60.254
        Ip_netmask  255.255.255.0
```

Aquí son evidentes los campos de dirección IP, ruteador y mascara de red. Sin embargo, al igual que en PCTCP, también hay valores que son específicos para Lan Work Place.

`Link support` indica el inicio de la sección de configuración del driver LSL. El driver LSL es un TSR que proporciona administración de buffers para el TCP/IP de Novell. Cabe mencionar que cada una de las secciones de este archivo son paralelas a los drivers que son cargados en el autoexec.bat. La línea de `buffers` especifica el número y el tamaño de los buffers utilizados para recibir paquetes. El ejemplo define 8 buffers de 1500 bytes cada uno, lo cual es suficiente para mantener 8 de los frames mas grandes de ethernet. `Mempool` es el número de bytes permitidos en el buffer de envío. El ejemplo muestra 4Kb para buffer de paquetes esperando por acuse de recibo.

La sección Link Driver contiene la configuración del adaptador de red. El nombre proporcionado en la linea de comando es el nombre del dispositivo controlador instalado en el autoexec.bat, en este caso es NE2000. `Int` se refiere al número de interrupción usado por la interfaz, `port` define la dirección de I/O usado por la interfaz.

La sección de `Frame` define el tipo de frame que se utilizará. Esta configuración en especifico no tiene nada que ver con TCP/IP, pero tiene que ser configurada correctamente para que la red trabaje. En el ejemplo, se especificó que se utilizarán frames tipo Ethernet_II (DIX). La mayoria de las redes TCP/IP y en particular aquellas con sistemas UNIX, usan este tipo de frame por default. Novell es una de las excepciones debido a que Netware de Novell utiliza frames IEEE 802.3 y utilizan este mismo tipo de frame par TCP/IP por default

La sección de `Protocol TCPIP` define la configuración estándar para TCP/IP. La línea de `Path TCP_CFG` es la ruta donde se encuentran los archivos de configuración de Lan Work Place. En esta ruta se encuentran los archivos hosts, resolv.cfg, networks protocols y services.

Ambos ejemplos son archivos mínimos que solo cubren las cosas requeridas al iniciar la configuración de TCP/IP. Hay muchos comandos de configuración que pueden agregarse en el archivo net.cfg y en el PCTCP.INI. Afortunadamente los valores por default son usualmente los correctos. La mayoría de los problemas de instalación se originan de valores de configuración incorrectos no de los valores proporcionados por el fabricante.

4.2. Windows en Red

Windows es una de las razones del continuo éxito de DOS. Windows no es un sistema operativo (hasta la versión 3.11), realmente es un GUI (graphical user interface) que corre sobre DOS como una aplicación. Para usar Windows, primero debe instalarse DOS. Los programas de Windows proporcionan una interfaz de usuario mas consistente que las aplicaciones de DOS. Casi cualquier persona que se sienta frente a una computadora con Windows encontrará su interfaz fácil de utilizar.

Windows extendió la vida de DOS al cubrir las carencias de DOS. Windows proporciona capacidad de manejo de hasta 16 Mbytes de memoria y soporta multitareas de forma limitada. Windows usa un esquema conocido como mutitarea cooperativa que recae en aplicaciones "bien portadas" para permitir al sistema operativo compartir recursos entre múltiples programas.

Todas las implementaciones de TCP/IP para dos son construidas en base a TSRs pero la configuración y sintaxis de los comandos de implantaciones DOS es muy amplia. Para windows es lo opuesto. La interfaz de usuario consistente reduce las diferencias en la configuración y uso de varias implantaciones, pero los desarrolladores de Windows tiene varias opciones de cómo construir el software. La discusión de la implantación de TCP/IP comienza con esas opciones de implantación.

Hay tres técnicas usadas para implantar TCP/IP sobre Windows: TSR, DLL o VXD.

- Implantación con TSR. Dado que Windows corre encima de DOS, la misma estructura de TSR utilizada para DOS se puede usar en Windows. Hay las mismas ventajas y desventajas. Un TSR es rápido, opera en tiempo real en respuesta a una interrupción. Sin embargo, usa memoria DOS preciosa, pero una de las razones de usar Windows es eliminar el limite de memoria de DOS. Una ventaja especial para Windows es que el TSR le puede dar servicio a todas las ventanas, incluso en la ventana de DOS.

- Dynamic Link Library (DLL). Un DLL es una librería que puede ser llamada por un programa sin haber sido ligada al programa en tiempo de compilación. Cuando me refiero a los programas, me refiero unicamente a programas Windows, ya que estos programas no pueden correr desde DOS, necesitan el ambiente Windows. Los DLL usan muy poca memoria y la que usan es la de Windows, no utilizan memoria DOS. Las implantaciones de TCP/IP basadas en DLL dependen completamente de Windows.

- Virtual Device Driver (VXD). VXD es un nuevo desarrollo para implantar TCP/Ip para Windows. Un VxD es un dispositivo controlador creado dentro de la maquina virtual de Windows. Como un dispositivo controlador de DOS, un VxD puede ser diseñado

para responder a interrupciones en tiempo real. Un VxD no utiliza memoria DOS y esta disponible para ventanas DOS, pero no para DOS cuando no esta Windows corriendo.

Los sistemas basados en TSR corren tanto en DOS como en Windows. Un TSR es deseable si lo importante es estandarizar el soporte a una implantación TCP/IP que corre en ambos ambientes.

Implantaciones basadas en DLL y VXD son nativas de sistemas Windows, los Vxd proporcionan un mejor desempeño que los DLL, ya que los Vxd son manejados por interrupciones. Muchos fabricantes han anunciado implantaciones VxD para la siguiente generación de software para Windows. La velocidad es lo importante para sistemas tales como servidores que son altamente utilizados y una PC puede ser utilizada como servidor cuando corre Windows 3.1. Sin embargo, la mayoría de las PC's no son muy sobrecargadas y la mayoría de los administradores de red utilizan sistemas operativos robustos como UNIX en los servidores.

No importando como sea implementado el software de TCP/IP, la cosa mas importante cuando se escoge un paquete TCP/IP para Windows son las aplicaciones que el paquete soporta y la calidad de esas aplicaciones. Hay muchos paquetes de TCP/IP para Windows de donde escoger. La mayoría de los paquetes de TCP/IP para DOS también funcionan en Windows. Ahora echemos un vistazo a los paquetes de TCP/IP para Windows mas utilizados.

El paquete de Microsoft es un stack de protocolos TCP/IP sin muchas aplicaciones y el paquete SPRY es un juego completo de aplicaciones sin un stack de protocolos. El paquete SPRY puede ser combinado con un paquete que incluye un stack, tal como el software de Distinct o de Microsoft.

Al parecer, puede parecer tonto proporcionar un conjunto de aplicaciones sin un stack de protocolos. ¿Y como pueden correr estas aplicaciones en stacks de distintos proveedores?. La respuesta está en el estándar winsock.

Winsock es un API estándar definido para correr TCP/IP sobre windows. Winsock viene de Windows Sockets, y es una implantación al estilo de los sockets TCP/IP de Berkeley para Windows. Las aplicaciones y stack de protocolos que cumplen con el estándar winsock deben ser capaces de interoperar.

Todos los paquetes antes mencionados cumplen con el estándar de winsock. Gracias a esto, se puede escoger el stack y las aplicaciones por separado de diferentes fabricantes.

4.2.1. Instalación de TCP/IP en Windows

Al igual que la instalación de TCP/IP en DOS, la instalación en Windows sigue dos pasos: copiar el software y modificar los archivos de configuración. La diferencia es que el sistema de Windows siempre utiliza un programa de instalación.

Es extremadamente importante tener toda la información de instalación a mano antes de comenzar con el programa de instalación. Por lo regular, sólo es cuestión de llenar los espacios en blanco en las cajas de dialogo que se van presentando.

A continuación presentaré un ejemplo de instalación de SuperTCP de Frontier Technologies:

El programa de configuración comienza desplegando una caja de dialogo que pide la información de la licencia. Después de esto, aparece una caja de dialogo en donde se puede elegir el tipo de instalación que se desea, personalizada o típica.

Posteriormente, el programa instala las aplicaciones que se eligieron y avisa de las modificaciones que se realizarán a los programas de arranque. Particularmente, este programa modifica el Autoexec.bat, config.sys y algunos archivos de configuración de Windows.

La mayoría de las implantaciones de TCP/IP realizan una copia de los archivos de inicio de DOS y de los archivos de configuración de Windows, sin embargo es conveniente realizar esta tarea antes de realizar cualquier instalación.

Después de que ha instalado las aplicaciones, el programa trata de encontrar si hay dispositivos controladores instalados. El adaptador de red debe ser instalado antes de instalar el software. Si no encuentra ningún controlador, el software pregunta por tipo de controlador que se desea instalar (NDIS, ODI, Packet, etc) y que tipo de hardware se tiene. El paquete incluye una selección de los controladores de adaptadores comunes; sin embargo, puede no tener el controlador exacto que se necesita, en tal caso, se puede seleccionar la opción de Other del menú de adaptador e introducir el controlador proporcionado por el fabricante.

Una de las partes con mas trucos es la configuración del controlador. Se deben de saber los detalles de la instalación de hardware para saber los parámetros de configuración del adaptador. Por lo regular, todas las tarjetas de red tienen software de configuración propia.

Una caja de configuración mínima aparece preguntando por la dirección IP, nombre de la PC, nombre del dominio y las direcciones para el servidor de nombres. Otras implantaciones pueden requerir cosas diferentes.

Una vez que la configuración mínima ha finalizado, se llama a un programa desde el cual se puede configurar la interfaz y algunas aplicaciones. No es necesario configurar todas la aplicaciones durante instalación inicial. Se recomienda no intentar configurar todas las aplicaciones al mismo tiempo, ya que esto puede causar confusiones y conflictos entre las mismas, también se recomienda terminar la instalación justo después de la configuración mínima. El tipo de información ingresada durante la configuración inicial es la misma de sistema a sistema. Una vez finalizada la instalación, en el administrador de programas aparece un grupo de programas para PCTCP.

Para entender porque los programas de TCP/IP modifican la estructura de los archivos de configuración de Windows, primero debemos saber cual es la función de ellos.

El archivo system.ini almacena la información de la configuración de hardware. El archivo Win.ini es un archivo que almacena una gran variedad de información de Windows. Este archivo es usado por algunas aplicaciones para almacenar su configuración.

El archivo Progman.ini tiene la información de configuración de los grupos del administrador de programas. Control.ini es el archivo que almacena los valores definidos por el usuario del panel de Control, tales como el esquema de color, el protector de pantalla, controladores instalados pro el usuario e información MIDI y multimedia. El archivo Mouse.ini almacena información relacionada al mouse.

Los archivos mas importantes de la configuración de Windows son el system.ini, win.ini y progman.ini. Los archivos INI tienen una estructura y sintaxis de comandos estándar. Los archivos estan divididos en secciones que comienzan con una etiqueta encerrada por paréntesis cuadrados. Por ejemplo, la sección de red dentro del archivo win.ini comienza con la etiqueta [network]. Cada comando de configuración esta escrito como una palabra clave, un signo de igualdad y alguna variable, por ejemplo: network.drv=c:\drivers\multi400.drv.

Las aplicaciones de Windows también utilizan archivos INI para retener su configuración. Como se habrá notado, algunas aplicaciones almacenan información en el Win.ini, pero otras crean archivos INI privados.

Hasta el momento hemos visto sólo ejemplos de computadoras que utilizan sólo una interfaz de red, sin embargo, habrá algunas que requieran mas de una, por ejemplo las computadoras portatiles. Las computadoras portátiles están conectadas a la red por medio de una interfaz ethernet y desde cualquier otra parte, se puede conectar por medio de un puerto COM corriendo SLIP o PPP.

4.2.2. Configurando SLIP

Los detalles varían, pero la mayoría de las implantaciones de TCP/IP proporcionan una técnica simple para configurar SLIP. El programa Chamaleon Custom demuestra que tan fácil es configurarlo.

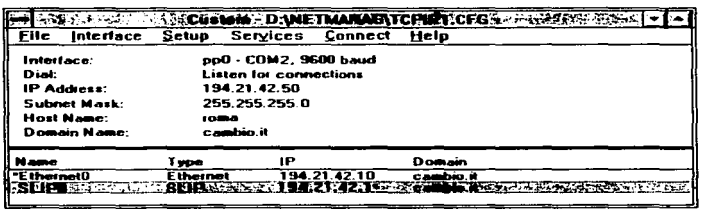


Figura 4.3 Programa Custom de Chamaleon

En el menú principal se debe seleccionar Interface / add. La figura 4.4 muestra la caja de dialogo para agregar interfaces.

Después se selecciona el tipo de interfaz de la lista. Hay varias opciones disponibles: SLIP (Serial Liner IP), PPP (point to point protocol) y CSLIP (Compresed Serial Line IP), todas ellas de tipo serial. SLIP es uno de los mas utilizados, CSLIP es una versión de SLIP con compresión de encabezado para mejorar el desempeño.

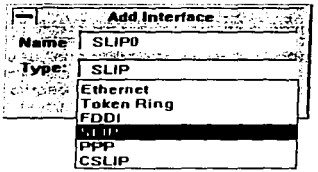


Figura 4.4. Agregando una interfaz de red en Chamaleon.

El sistema genera el nombre de la interfaz en base al tipo de interfaz, cada una se va numerando desde 0.

Una vez instalada la interfaz, se necesita configurar. Algunos parámetros son asignados por default, pero algunos necesitan proporcionarse manualmente. Por ejemplo, se debe configurar el puerto COM. En la figura 4.5 se muestra un ejemplo de la configuración de los puertos.

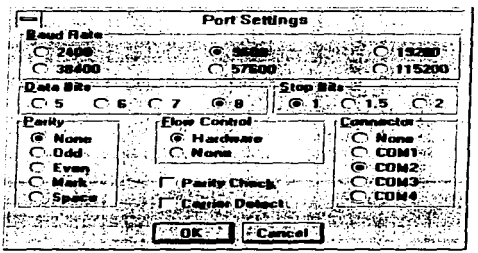


Figura 4.5 parámetros de configuración del puerto de comunicaciones

Todos estos valores de configuración deben coincidir con los valores del modem y deben ser compatibles con SLIP. Los valores estándar par SLIP son 8 bits de datos sin paridad. Nunca se debe utilizar control de flujo por software en una linea SLIP. Si se requiere de control de flujo, se debe utilizar el control por hardware.

La configuración del modem define los comandos usados por el modem. La cadena de comandos del modem se utilizan para inicializar, marcar el número y colgar la linea. Estos comandos son claves para lograr una comunicación. Chamaleon proporciona valores por default para modems Hayes, Telebit y Multitech. Es importante conocer los comandos del modem que ser requiere instalar.

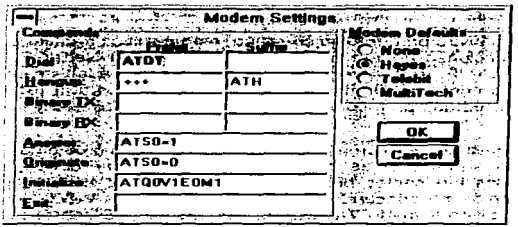


Figura 4.6 Configuración de los parámetros del Modem

La opción de login define el nombre de usuario y el password utilizado para tener acceso al servidor de SLIP. También permite definir un comando de inicio,esto se utiliza para servidores que soportan multiples protocolos y que requieren iniciar el tipo de protocolo a utilizar.

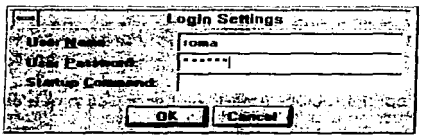


Figura 4.7. Caja de dialogo de inicio de sesión.

Estos son todos los parámetros comúnmente utilizados para configurar el comportamiento de una sesión de SLIP.

4.2.3. Configuración de Windows para Trabajo en Grupos.

Windows para trabajo en grupos (WfW, Windows for WorkGroups) es una variante de Windows 3.1 que reconoce su habilidad de actuar como un cliente o servidor de red. Desde la perspectiva del usuario, WfW y Windows son identicos; WfW corre software de Windows y si no se esta interesado en trabajo con redes, WfW puede tomar el lugar de Windows sin tener que realizar ningún cambio.

Sin embargo, WfW es software para trabajo en grupo, esto significa trabajar con redes. WfW tiene sus propias aplicaciones de red basadas en el protocolo NetBIOS. NetBIOS puede correr sobre una red TCP/IP. WfW agrega el archivo Protocol.ini a la lista de archivos de inicialización. Este archivo contiene información de configuración para la red de WfW. Describe los adaptadores de red, los protocolos de transporte, controladores y los enlaces entre todos ellos.

Los fabricantes de software TCP/IP comercial tienen presentes los problemas de correr múltiples redes en un solo sistema y las dificultades de configurar correctamente el Protocol.ini. La mayoría de los fabricantes proporcionan herramientas en el programa de configuración para instalar en forma adecuada TCP/IP en una sistema WfW. Por ejemplo, NetManage proporciona un folleto llamado WorkGroup ChamaleonNFS con el software de Windows. El folleto dice como instalar su producto en un ambiente WfW. Básicamente, dice que primero se debe configurar y probar la red NetBIOS de WfW y posteriormente instalar el software de Chamaleon como se hace en cualquier Windows. El programa de configuración detecta la presencia de WfW, instala los componentes necesarios y hace las modificaciones necesarias para ser compatible con él.

Otro punto que se debe tomar en cuenta es que se debe instalar el stack de TCP/IP que es compatible con WfW y se basa en winsock para correr las aplicaciones que se deseen. Este stack esta disponible vía ftp anónimo en ftp.microsoft.com y esta localizado en el directorio peropsys/windows/public/tcpip, el archivo que contiene el stack se llama wfwt32.exe. Este programa se debe copiar en un directorio y una vez ahí, se ejecuta. Este programa es auto-descompactable. Una vez finalizada la ejecución aparecerán aproximadamente 33 archivos.

WfW tiene un programa de configuración llamado Network Setup que esta localizado en el grupo de programas de red. Para instalar TCP/IP debe iniciarse el programa Configuracion de red y seleccionar Controladores de la ventana que se despliega. En la ventana de Controladores se presiona el botón de agregar protocolo. Despues de presionar el botón aparece una ventana que muestra los protocolos disponibles, por lo regular, el stack de TCP/IP no aparece. Para agregarlo, se escoge la opcion de protocolo no listado y en la ruta se escribe la ruta en la que se descompactó el programa wfwt32.exe. Cuando se presiona el boton de aceptar, automáticamente se instala el software.

Una vez finalizada la instalación, se procede a configurar el software. En la figura 4.8 se muestra la ventana de configuración de TCP/IP. Al igual que con la mayoría del software de configuración, TCP/IP para WfW también necesita la dirección IP, la mascara, el ruteador por default, los servidores de nombres, el nombre de la computadora y el dominio al que pertenece.

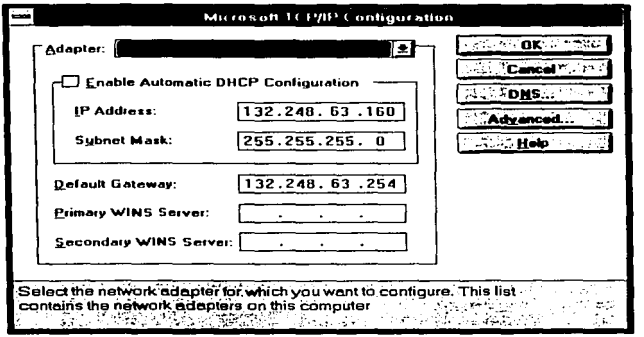


Figura 4.8 Configuración del stack de TCP/IP para WfW

4.2.4. TCP/IP para Windows 95

La nueva generación de Windows no es dependiente de DOS. Windows 95 es un sistema operativo en sí, ya no se ejecuta sobre DOS. Windows 95 es un sistema operativo completamente integrado. La PC realiza el boot directamente sobre Windows 95, proporciona todo el soporte de E/S para aplicaciones DOS que corren desde Windows 95.

Proporciona una nueva interfaz de usuario completamente nueva. Ya no existen mas el administrador de programas ni el administrador de archivos, todo esto ha sido reemplazado por el explorador.

Windows 95 proporciona soporte para redes integrado. Incluye los protocolos TCp/IP, Netware y NetBEUI. Tiene un uso extensivo de código de 32 bits. El kernel es código de 32 bits protegido, el cual proporciona gran confiabilidad. La mayor parte del sistema operativo esta construido en base a código de 32 bits. Sólo el código que requiere compatibilidad con versiones anteriores de Windows usan código de 16 bits.

Otra de las características importantes de Windows 95 es el soporte a multitareas real. Las aplicaciones de 32 bits podrán tomar ventaja de esta característica, pero las aplicaciones de 16 bits tendrán todavía que utilizar multitareas cooperativas.

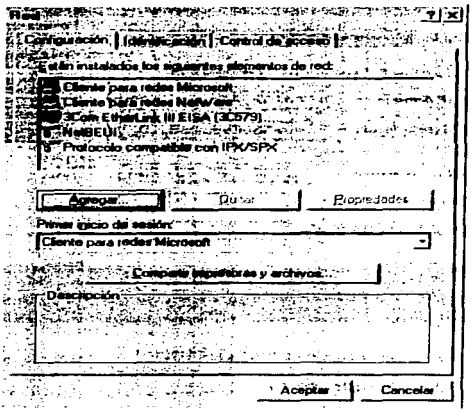


Figura 4.9.Ventana de configuración de Red.

La administración del sistema se ha mejorado ya que Windows 95 incluye nuevas herramientas para el manejo del sistema y de los usuarios. Estas herramientas pueden ser utilizadas en forma remota, por lo que puede ser administrada en forma remota. Todos los parámetros de configuración están centralizados en una base de datos conocida como registro del sistema. El soporte Plug-And-Play esta interconstruido con el sistema operativo para simplificar la configuración del hardware.

Para configurar TCP/IP en una red de Windows 95, primero se debe entrar al Panel de Control. Una vez ahí, se selecciona la opción de red, tal y como se realizaba en Windows 3.1. La figura 4.9 muestra la ventana de configuración de red.

En la pestaña de configuración, el botón de compartir archivos e impresoras muestra una caja de dialogo como la de la figura 4.10. El cuadro donde aparece primer inicio de sesión define que servidor debe de validar el acceso al sistema. Cuando se inicia Windows 95, el sistema pide un nombre de usuario y un password.

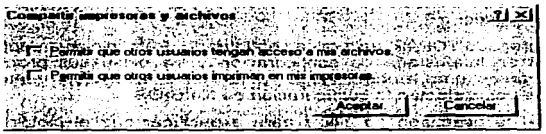


Figura 4.10 Opciones de compartir impresoras y archivos

El botón de Agregar abre una caja de dialogo en la que se debe de seleccionar lo que se quiera agregar, las opciones disponibles son: Adaptador, protocolo, cliente y servicio. Los adaptadores, como su nombre lo indica sirve para agregar un nuevo adaptador de red. La opción de protocolo agrega un nuevo protocolo, por ejemplo IPX/SPX, entre otros. La opción de clientes proporciona la facilidad de que Windows 95 se convierta en un cliente de otras redes, como por ejemplo, cliente de Novell Netware. La opción de agregar servicios, permite agregar servicios como el de compartir archivos e impresoras para redes Novell o Microsoft.

La pestaña de Identificación es usada para configurar el nombre de NetBIOS y el grupo de trabajo que utilizará la máquina.

La pestaña de control de acceso es utilizada para seleccionar cuando debe utilizarse control de acceso compartido como en WfW o control acceso por niveles como Windows NT.

Para instalar TCP/IP es necesario seleccionar el botón de agregar dentro de la pestaña de configuración. Una vez ahí, se debe seleccionar protocolo. Una vez seleccionada la opción de protocolo y presionar el botón de agregar, aparece una caja de dialogo similar a la de la figura 4.11.

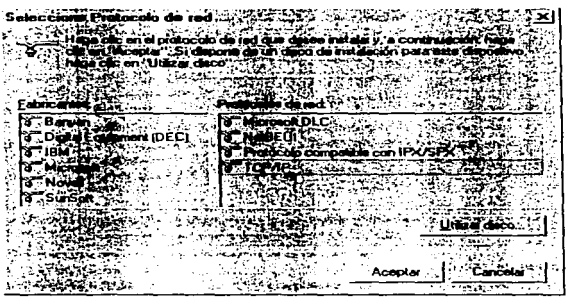


Figura 4.11 Ventana de protocolos.

En la ventana de seleccionar protocolo de red, se selecciona Microsoft dentro de las opciones de fabricantes.

Una vez agregado TCP/IP, solo resta configurarlo. Para iniciar la configuración, se tiene que seleccionar en la pestaña de configuración y se piden sus propiedades. Al elegir propiedades, aparece una ventana similar a la figura 4.12. Esta ventana contiene seis pestañas, tres de estas afectan la configuración de NetBIOS.

La pestaña de avanzado se utiliza para especificar que el protocolo por default es TCP/IP. La pestaña de enlaces se utiliza para identificar los componentes de red que se comunican por medio de TCP/IP Si se utilizará encapsulamiento de NetBIOS dentro de TCP/IP se debe asegurar que todos los componentes de red de Microsoft listados en esta pestaña estén seleccionados. La pestaña de configuración de WINS permite habilitar o deshabilitar Wins. Wins es un servidor de nombres para nombre de NetBIOS. Generalmente se deshabilita cuando se utiliza un DNS en una red TCP/IP.

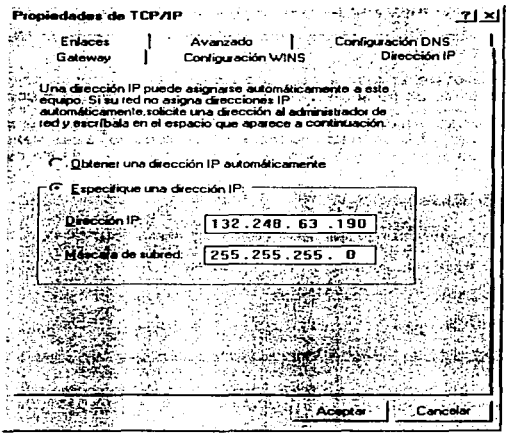


Figura 4.12 Propiedades de TCP/IP.

Las otras tres pestañas, Dirección IP, Gateway y Configuración DNS son críticas para la configuración de TCP/IP. La pestaña de Dirección IP se muestra en la figura 4.12. y se utiliza para introducir la dirección IP de la máquina.

La pestaña de Gateway se utiliza para agregar el gateway por default. La pestaña de configuración del DNS se utiliza para agregar los servidores de nombres disponibles. Aquí también se define el nombre del host y el dominio al que pertenece.

4.2.5. Redes TCP/IP en Windows NT

Los servidores de red necesitan sistemas operativos confiables y robustos. DOS y Windows 3.1 no son opciones. La mayoría de los administradores de redes de área local toman la opción de Netware y los administradores de redes TCP/IP toma a UNIX para crear servidores de red confiables. Naturalmente Microsoft quiere cambiar esto por medio de Windows NT.

Windows 4.0 usa la misma interfaz gráfica de Windows 95, pero es mucho más que un GUI. NT es un sistema operativo multitareas, multi-usuario y multi-hilos. Windows NT viene con versiones para un solo usuario para estaciones de trabajo de alto rendimiento y en versión de servidor de red con múltiples usuarios.

NT esta diseñado para ser un sistema operativo con capacidades de red. Desde las primeras versiones, Windows NT incluye el software de TCP/IP y basa su red empresarial en TCP/IP. Sin embargo, TCP/IP no es el único protocolo que de red en un sistema NT.

La mayoría de las aplicaciones de red ligadas directamente con Windows NT están basadas en NetBIOS. Para entender como realiza el trabajo de red, incluyendo TCP/IP, debemos entender primero NetBIOS.

NetBIOS y el protocolos asociado NetBEUI (NetBIOS Extended User Interface) tienen amplias bases en la estrategia de trabajo de redes de Microsoft. Los productos Lan Manager, Windows for WorkGroups, Windows 95 y Windows NT están basados en estos protocolos.

BIOS, el sistema básico de entrada y salida es la parte de DOS que define las llamadas a la entrada y salida de los dispositivos. NetBIOS es una extensión de esto. NetBIOS controla las llamadas de entrada y salida relacionadas con la red. Originalmente NetBios se implanto en una ROM dentro de la tarjeta de interfaz de red de Sytek. Este era un protocolo monolítico, el cual tomaba los datos de la aplicación y los enviaba directamente a la red. Hoy en día NetBIOS es un API que define como los programas de aplicación deben de realizar las peticiones de servicio de las redes subyacentes.

IBM y Microsoft han agregado funciones a la definición básica de NetBIOS durante años para producir la versión actual de NetBEUI. No hay que confundir NetBEUI con NetBIOS, NetBEUI incluye el API de NetBIOS, el protocolo de servicio de bloques de mensajes (SMB) y el protocolo de frames de NetBIOS (NBF). SMB como NetBIOS es un API que define como las aplicaciones hacen peticiones para los servicios de red, pero NetBEUI es más que un API. También incluye el protocolo NBF que construye frames de NetBIOS para transmisión sobre la red.

NetBIOS requiere un poco de memoria y corre en cualquier tipo de PC, es rápido, ligero y esta disponible para redes LAN. Sin embargo, NetBIOS es solo para aplicaciones de redes de área local. NetBIOS no puede ser utilizado para redes WAN o en redes de gran tamaño debido a que es un protocolo no ruteable y depende de el medio subyacente de broadcast. El término de no ruteable se refiere a que el protocolo puede pasar a través de los ruteadores. Los paquetes de NetBIOS solo pueden ser pasados en una red física. No tiene protocolo de ruteo no tiene una estructura de direcciones independiente. Depende completamente de las direcciones de capa física, lo cual lo limita a una sola red física. Se dice que es dependiente del broadcast debido a que utiliza el broadcast físico, y si la red no puede proporcionarlo, simplemente el protocolo no funciona.

La forma en como se agrega un nuevo nodo dentro de una red de Windows para trabajo en grupos es un buen ejemplo de la dependencia de NetBIOS con el broadcast físico. Cuando WfW comienza envía un broadcast con un paquete de petición registro de nombre. El paquete contiene el nombre de NetBIOS propuesto que identifica al sistema. Si otra computadora en la red tiene el mismo nombre, esta responde al broadcast con un paquete de respuesta negativa de registro de nombre. Si el nuevo nodo no recibe la respuesta negativa a su broadcast, esta utiliza el nombre como su identificador.

El nombre es literalmente utilizado como la dirección del nodo. Los campos de origen y destino de un frame NetBIOS son cada uno de 16 bytes de longitud. Los campos contienen los nombres de origen y destino de las computadoras.

El esquema de nombres tiene la ventaja de ser intuitivo, ya que las personas prefieren identificar las cosas por nombres que por números. Además, no se requiere que exista un servidor central que realice la autentificación.

Una de las desventajas de este esquema es que cada nombre debe ser enviado en broadcast a cada nodo en la red, lo cual es una dificultad en redes de gran tamaño. En algunas ocasiones, con redes muy grandes es posible que no se puedan mantener únicos los nombres y dado que no hay una autoridad centralizada que controle los nombres, no se descarta la posibilidad de repetición de nombres.

NetBIOS ha cambiado y ha pasado de ser un protocolo monolítico a un protocolo estratificado. Este cambio provoca confusión en la forma en que el término "NetBIOS" es usado, ya que algunas veces se refiere a todo el protocolo y alguna veces sólo a la interfaz de aplicación. Sin embargo, el efecto real de este cambio es que en la forma actual de interfaz de aplicaciones, NetBIOS no depende del protocolo subyacente NBF. NetBIOS puede correr en una variedad de protocolos diferentes de red, incluyendo TCP/IP. Instalar TCP/IP no elimina la disponibilidad de aplicaciones basadas en NetBIOS correr en redes grandes que incluyen ruteadores. Se puede realizar esto encapsulando mensajes NetBIOS dentro de datagramas de TCP/IP. El protocolo que hace esto posible es NetBIOS over TCP/IP (NBT).

NetBIOS over TCP/IP es un protocolo estándar definido en el RFC 1001 y 1002. La versión de Microsoft para NBT se basa en estos estándares. El protocolo NBT de Microsoft esta basado en la arquitectura de b-node definido en el RFC. Un b-node (broadcast node) es un nodo final que utiliza mensajes de broadcast para registrar su nombre y para pedir los nombres a otros sistemas en la red.

Los mensajes de NetBIOS son encapsulados en mensajes de UDP y enviados utilizando direcciones IP de broadcast. En efecto, IP actua como el medio de broadcast para el protocolo NetBIOS.

La arquitectura de b-node no elimina el problema de la dependencia de broadcast, asi que Microsoft utiliza una arquitectura b-node modificada. NBT de Microsoft carga un cache con mapeos de nombres de NetBIOS a direcciones IP del archivo LMHOSTS. El cache resuelve el problema de del uso de broadcast para la resolución de nombres.

Windows NT esta diseñado para ser un sistema operativo orientado a redes. Durante la instalación del sistema operativo, el programa de configuración solicita la información del adaptador de red. Una vez que se decide instalar Windows NT, se debe tener presente la información de la tarjeta de interfaz de red, el software que se quiere utilizar, la información especifica para TCP/IP y la configuración especifica de NetBIOS.

El procedimiento para configurar un adaptador de red es muy parecido al seguido para Windows 95.

Primero se selecciona una tarjeta de red de la lista de adaptadores. Una vez seleccionada, se procede a configurar los parámetros que se han venido mencionado desde el principio del capitulo, IRQ y dirección de puerto de E/S.

Una vez que se tiene el adaptador de red instalado, el siguiente paso es configurar el software. Para agregar TCP/IP, se selecciona la opción de agregar software y entonces se selecciona la opción de Protocolo TCP/IP y componentes relacionados de la lista editable. AL seleccionar TCP/IP, se muestran las opciones que aparecen en la figura 4.15.

Esta ventana lista todos los componentes disponibles en el sistema. La ventana de la figura 4.15 es de un servidor NT, se muestra esta porque hay menos opciones de software para las estaciones de trabajo. Por default el sistema instala trabajo con TCP/IP cuando se presiona el botón de continuar. Esto proporciona el stack de protocolos TCP/IP completamente funcional. Para instalar un sistema básico con herramientas tales como el PING, Telnet y FTP, se debe elegir la opción de utilerias de conectividad.

Una vez que se ha instalado el software, es preciso configurarlo. Toda la configuración de TCP/IP es muy parecida a la configuración en Windows 95.

La configuración de NBT combina algunos aspectos de la configuración estándar de NetBIOS con opciones propias de NBT. Los parámetros tradicionales de NetBIOS, tales como nombre del host y el grupo de trabajo son proporcionados al inicio de la instalación, cuando NT identifica que se ha instalado un adaptador de red automáticamente instala el software NetBEUI.

El nombre de NETBIOS es simplemente un nombre de longitud menor a 15 caracteres. El uso de mayúsculas y minúsculas es indistinto ya que los caracteres que se introducen son interpretados automáticamente y convertidos a mayúsculas. Se necesita que el nombre sea único entre el host NetBIOS y los sistemas con los que se comunicará. Me refiero a los host con los que se comunicará para poner énfasis en que debe ser único tanto para aquellos host a los que pueda tener acceso directo de broadcast y aquellos con los que se puede comunicar gracias al archivo lmhost.

El nombre del grupo de trabajo no se ha explicado aún. Un grupo de trabajo es una agrupación jerárquica de hosts, parecida a la estructura de directorios. Los grupos de trabajo organizan los recursos de red de la misma forma que los directorios organizan los recursos de archivos. Esta agrupación tiene la ventaja de que al agrupar en grupos, los usuarios tienen mayor facilidad para la ubicación de los recursos.

Los grupos de trabajo no están diseñados para el manejo de seguridad. El grupo de trabajo no limita a los hosts que no están dentro del grupo de trabajo la compartición de recursos. La seguridad se refuerza utilizando autentificación de usuario con password.

Cada sistema NetBIOS espera unirse a algún grupo, es decir, el sistema espera tener configurado el nombre del grupo de trabajo. Este nombre de grupo de trabajo debe ser asignado por el administrador de la red. Si no se proporciona el nombre del grupo, el sistema automaticamente le asigna el de workgroup.

El nombre del grupo de trabajo puede ser modificado en la ventana de configuración de red. Para cambiarlo, simplemente se presiona el botón de modificar y se cambia el nombre del grupo de trabajo.

No hay que confundir un grupo de trabajo con un dominio. Un grupo de trabajo para NT es exactamente lo mismo que para Windows 3.11, simplemente una forma de organizar la variedad de sistemas dentro de una red NetBIOS. Un dominio NT es un dominio NT administrado por un servidor NT.

No se puede simplemente unirse a un domino NT. El sistema debe ser agregado al dominio por el administrador del mismo.

Hasta el momento sólo he mencionado el archivo lmhost. Este archivo nos permite realizar conexiones de NetBIOS con computadoras en otras redes.

Este archivo es muy parecido al archivo /etc/hosts de UNIX y funciona de manera similar. La diferencia es que el archivo LMHOST realiza el mapeo entre nombres de NetBIOS y direcciones IP. Un ejemplo de un archivo LMHOSTS muestra cuan parecidos pueden ser estos archivos:

```
132.248.52.159      creta
132.248.56.200      quetzal
200.15.45.156       centra
```

Cada entrada en el archivo LMHOSTS contiene una dirección IP separada por espacios en blanco del nombre de NetBIOS asociado con esa dirección. Una entrada no debe exceder una línea y los comandos inician con el símbolo #.

El archivo LMHOSTS es almacenado en el directorio de Windows en los sistemas WfW y en el directorio Winnt\system32\drivers\etc de Windows NT.

Se puede utilizar el mismo archivo para WfW y Windows NT, sin embargo, el archivo LMHOSTS de NT tiene algunas opciones que no son soportadas por WfW. Estas opciones comienzan con un # por lo que WfW lo ve solo como comentarios. Estas opciones son:

#PRE Esta opción origina que la entrada se precargada en el cache y retenida permanentemente ahí. Normalmente las entradas son solo enviadas al cache cuando son utilizada para la resolución de nombres y son retenida en el cache sólo unos minutos. Esta opción puede ser utilizada para agilizar la resolución de nombres comúnmente utilizados.

#DOM:dominio Identifica los controladores del dominio NT. La variable dominio es el nombre del dominio NT para el cual este sistema es un controlador.

#INCLUDE archivo. Especifica un archivo remoto que debe ser incorporado en el archivo LMHOSTS local.

La configuración de los servicios de TCP/IP para Windows NT es similar a la configuración en Windows 95. Sin embargo, Windows NT presenta mas opciones para la configuración de servicios. La figura 4.13 muestra las opciones de configuración para redes en Windows NT.

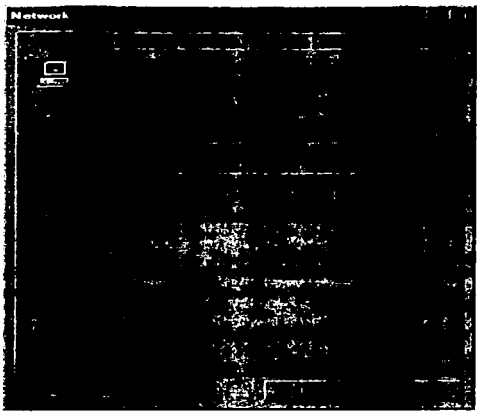


Figura 4.13 Opciones de configuración de Windows NT

La figura 4.13 muestra a simple vista el nombre de la computadora y el dominio al que pertenece. La pestaña de servicios permite establecer la configuración para cada uno de los servicios disponibles. La figura 4.14 muestra los servicios disponibles.

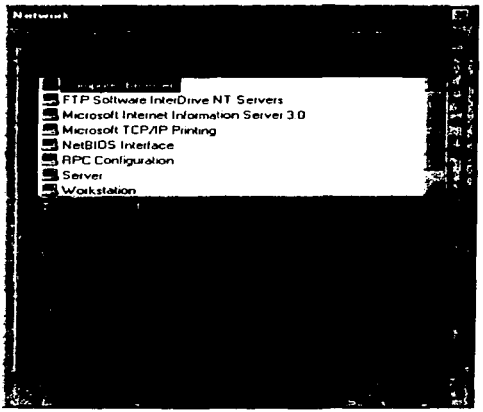


Figura 4.14. Servicios disponibles para Windows NT

Cada uno de los servicios tienen sus propias opciones. Por lo regular, las configuraciones de cada uno de los servicios especifican la forma de arranque. Las opciones son manual, automatica o deshabilitada.

Al igual que en Windows 95, la pestaña de protocolos incluye los protocolos básicos, NetBEUI, NetBIOS y TCP/IP. La configuración de TCP/IP presenta algunas opciones adicionales, tales como la de ruteo. Esta opción le permite a la computadora actuar como un ruteador. La pestaña de DHP relay permite configurar los servicios de DHP para computadoras sin disco que desean obtener su dirección IP, los servidores DHP también pueden actuar como servidores de BOOTSTRAP. El protocolo BOOTSTRP sirve para configurar la inicialización remota, esto es especialmente funcional para máquinas sin disco que requieren obtener todo el sistema operativo de algún servidor.

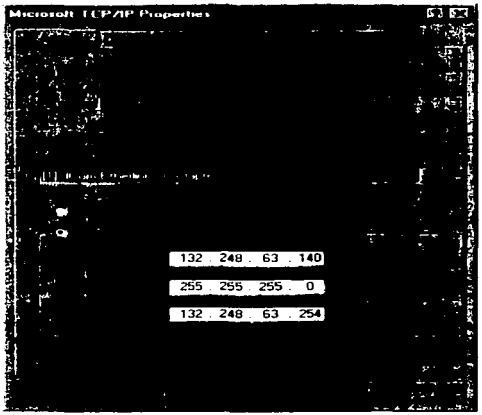


Figura 4.15 Configuración de TCP/IP en Windows NT

Por último examinaré la configuración de los servicios de red para MacIntosh. MacIntosh tiene acceso a la red por medio de un programa llamado MacTCP. Este programa ya está incluido en las nuevas versiones. La figura 4.16 muestra la primera pantalla, en esta observamos dos opciones: Appletalk y FreePPP. La opción de AppleTalk determina que la computadora utilizará el protocolo Appletalk para enlazarse con otras computadoras. FreePP indica que la conexión se hará en base a PPP. Hay otras opciones que se pueden agregar dependiendo el tipo de enlace que se tenga.

Por ejemplo, si se tiene una interfaz de Ethernet, en esta pantalla aparecería ethernet en lugar de freePPP.

La ventaja de los equipos MacIntosh es que no hay necesidad de configurar valores tales como IRQ, dirección de memoria base,etc. Automáticamente el sistema operativo asigna los valores.

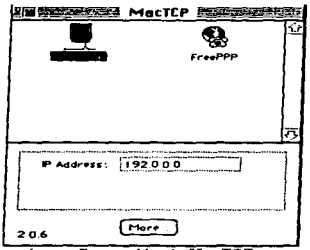


Figura 4.16. Opciones de configuración de MacTCP

Para configurar los valores de TCP/IP, aparece una ventana de dialogo como la de la figura 4.17. Los valores que se presentan aquí son bastante parecidos a los presentados en los programas anteriores.

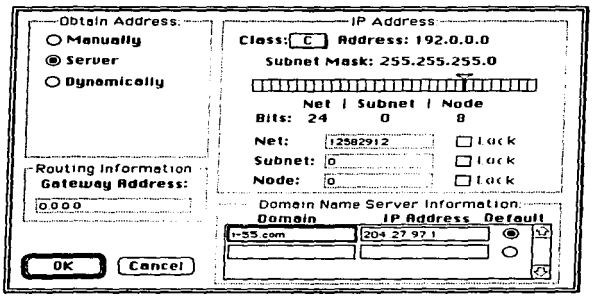
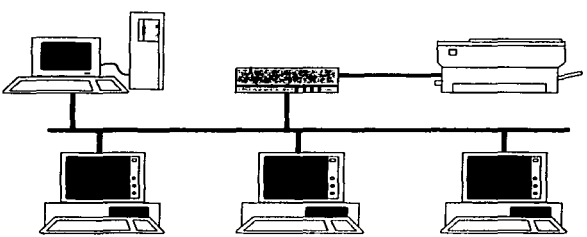


Figura 4.17. Configuración de los servicios de TCP/IP en MacTCP

5

La red de la Facultad de Química



Lista de Figuras

Historia de la Facultad de Química.

En enero de 1913, el Químico Juan Salvador Agraz presenta la primera iniciativa para preparar profesionales y maestros del área de la Química al presidente Madero. En 1914 realiza otro intento con José Vasconcelos, en aquel entonces Secretario de Instrucción Pública y Bellas Artes

El primero de octubre de 1915 el Químico Juan Salvador Agraz logra convencer a don Félix E. Palavicini y el 21 de diciembre de 1915 es designado director fundador de la primera Escuela de Química del país, la cual no contaba en esos momentos con local, ni aparatos, ni alumnos, ni maestros. Agraz era ya director de lo que aún tenía que hacerse.

Gracias a los esfuerzos del Químico Agraz se abrieron las carreras de: Químico Industrial, Perito en industrias y Práctico en industrias. El 3 de abril de 1916 iniciaron sus estudios 40 alumnos y 30 alumnas sin ceremonia alguna, en el local asignado a la Escuela en el pueblo de Tacuba. El 23 de septiembre de 1916 se hizo la solemne inauguración.

El 31 de enero de 1917, en la Cámara de Diputados se aprueba la supresión de la Secretaría de Instrucción Pública y Bellas Artes, y la Escuela de Química para así depender temporalmente del Gobierno del Distrito Federal, mientras la Universidad lo haría del Departamento Universitario, dependiente a su vez del Poder Ejecutivo Federal.

El 5 de febrero de 1917, la Escuela Nacional de Química Industrial fue incorporada a la Universidad gracias al apoyo del rector Macías y a las gestiones de Agraz ante la Cámara de Diputados.

Para 1949 la Escuela Nacional de Ciencias Químicas contaba con cuatro carreras: Ingeniería Química, Químico, Químico Farmacéutico Biólogo y Químico Metalurgista.

En el año de 1950, el 5 de junio se colocó la primera piedra de la que iba a ser Ciudad Universitaria , y es así como se inician los últimos años de la Escuela de Tacuba.

En la actualidad la Facultad de Química consta de 10 edificios, 4 de ellos ubicados a un costado de la Facultad de Ingeniería y 2 Complejos ubicados cerca del metro Universidad. Los dos complejos constan en conjunto de 6 edificios.

La red de la UNAM (RedUNAM)

La red de la UNAM al igual que las redes de otras universidades surge de la necesidad de comunicar a los investigadores de distintas áreas. En los inicios de los 70's se utilizan las instalaciones telefónicas para conectar computadoras a una computadora central. Esta técnica tiene gran aceptación en los círculos informáticos y se difunde no sólo dentro de la Universidad, sino que también comienza a difundirse a otras áreas de país. Su utilización primaria consistía de conexiones de terminales de caracteres, de graficación e impresión y conexión de estaciones de trabajo.

En 1987, la UNAM establece la primera conexión a la red BITNET haciendo uso de un enlace telefónico desde Ciudad Universitaria hasta el Instituto Tecnológico de Estudios Superiores de Monterrey y de ahí a San Antonio Texas.

En el año de 1989 se estableció un enlace satelital entre el Instituto de Astronomía y la red NSF de Estados Unidos. También en este año se realiza una conexión entre el Instituto de Astronomía y la Dirección General de Servicios de Cómputo Académico utilizando fibra óptica.

En ese momento se comenzó a gestar lo que llegaría a ser la REDUNAM. Las dependencias del subsistema de la investigación comenzaron con la adquisición masiva de computadoras personales y la interconexión de las mismas. Esto permitió el desarrollo de una infraestructura de comunicaciones con fibra óptica.

En forma paralela, se establecieron enlaces satelitales hacia Cuernavaca, Mor., y San Pedro Mártir en Ensenada, Baja California Norte,. También se estableció el primer enlace de microondas de alta velocidad entre la Torre II de Humanidades y la Dirección General de Servicios de Cómputo Académico, DGSCA, sobre la Ciudad de México.

En 1990 la UNAM se incorpora a la red Internet, con lo que se convierte en la primera institución en Latinoamérica en conectarse a esta red que enlaza a millones de máquinas en todo el mundo.

Para 1991, se incorporan los servicios de Telnet, FTP, correo electrónico y listas de correo, En 1992 se instaura el servicio de gopher y en 1993 bases de datos y revistas electrónicas. Los servicios de videoconferencia, WWW y traducción español-ingles se pusieron en marcha en 1995.

Actualmente RedUNAM tiene una infraestructura instalada para más de 170 redes locales de cómputo. La Red enlaza a cerca de 8000 computadoras en la UNAM entre sí y alrededor de un millón de computadoras en el resto del mundo. Sin embargo, RedUNAM no sólo realiza transmisiones de datos, también realiza transmisiones de voz y datos.

Hoy en día, alrededor del 90% de los miembros de nivel licenciatura, posgrado e investigación están integrados a la red. El sistema es descentralizado redundante y esta integrado por 31 Nodos de Cómputo y Telecomunicaciones enlazados entre sí por fibra óptica.

La estructura principal de REDUNAM es un anillo de FDDI (una fibra óptica activa y una de respaldo que pueden transportar información hasta 100 Mbps y que enlaza a 5 ruteadores principales). La figura 5.1 muestra la estructura del BackBone principal de la UNAM.

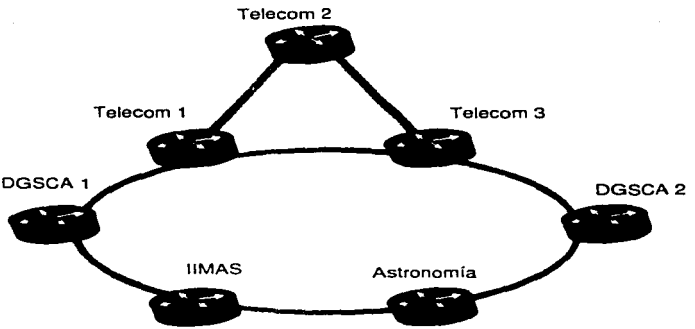


Figura 5.1 Backbone Principal de RedUNAM

Con respecto a los servicios de red, la Facultad de Química consta actualmente de 4 segmentos de red. Cada uno de estos segmentos es capaz de albergar teóricamente 255 máquinas, sin embargo, como ya hemos visto en los capítulos anteriores, hay direcciones IP reservadas, tales como la dirección de red, la de broadcast, la dirección del ruteador por default y otras relacionadas con la administración local de la red. En esta tesis sólo contemplaré la implantación de los servicios de red en 3 de ellos, ya que el restante es de nueva creación y, hasta la fecha de término de esta tesis, no se han implantado servicios en forma en dicho segmento.

La Facultad de Química ha formado un patronato de Cómputo, el cual se encarga de controlar las actividades de cómputo. El Centro de Informática se encarga actualmente de realizar la gestión de la red.

La Facultad de Química consta de varios edificios, 4 de ellos ubicados a un costado de la Facultad de Ingeniería y 2 Complejos ubicados cerca del metro Universidad. Los dos complejos constan en conjunto de 6 edificios.

Para darnos una idea del tamaño de la red de la Facultad, reunamos todos los datos: hay un total teórico de 1000 direcciones IP distribuidas en 10 edificios. La labor de dar mantenimiento a toda esta infraestructura es una cuestión compleja, si se toma en cuenta la poca disponibilidad de personal.

La ubicación de la red de la Facultad con respecto a RedUNAM se muestra en la figura 5.2.

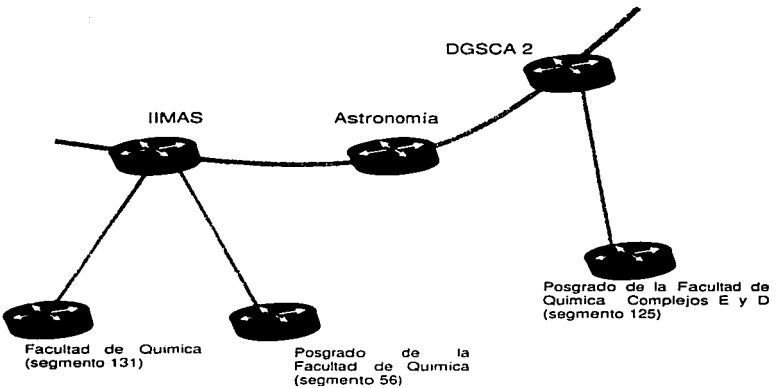


Figura 5.2 Ubicación de la red de la Facultad de Quimica en la RedUNAM.

5.1. Implantación en el segmento 131

Una vez que he mostrado la ubicación de la red de la Facultad de Química en el contexto de RedUnam, toca ahora mostrar la estructura física de la red en cada uno de los edificios. Todo el trabajo aquí descrito es sólo un extracto, ya que los mapas originales contienen información detallada de cada uno de los nodos de la red como son: número de serie, numero de inventario, modelo, tipo de procesador, dirección IP, puerto del concentrador al que va conectado y responsable.

El mapa de la ubicación de los edificios que se encuentran a un costado de la Facultad de Ingeniería se muestra en la figura 5.3. A primera vista, el mapa de la figura no muestra gran información, sin embargo, analizándolo se puede descubrir que el paso de señal de un edificio a otro no es tarea sencilla. Para realizar el enlace entre edificios se requirió de la instalación de Fibra óptica. Esta elección se debió principalmente a la gran cantidad de equipo industrial que existe en el edificio de Ingeniería Química.

El cable coaxial no representaba una solución viable, ya que la alta cantidad de ruido eléctrico presente en el ambiente, hacían que la transmisión de datos no fuera confiable ni

rápida. La baja en las velocidades de transmisión dentro de un ambiente ruidoso se debe principalmente a la gran cantidad de paquetes perdidos que requieren ser transmitidos de nueva cuenta.

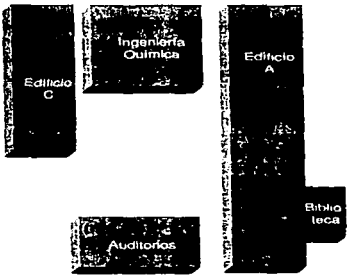


Figura 5.3. Esquema de ubicación de los edificios del Segmento 131

5.1.1. Edificio C

En este segmento se implantaron cerca de 130 servicios de TCP/IP para equipos personales. La mayor parte de las implantaciones se realizaron en el laboratorio de cómputo del edificio C.

En el laboratorio de cómputo se implantó TCP/IP en Windows para Trabajo en Grupo. En el laboratorio se presentó una dificultad bastante común en algunos centros de cómputo.

Cuando se hace la asignación de la dirección de memoria, IRQ y dirección de puerto de entrada y salida en Windows 3.11, estos valores no son directamente actualizados en la tarjeta de red.

Cuando WfW trata de inicializar la tarjeta adaptadora de red, los valores de la tarjeta y los valores en el archivo PROTOCOL.INI no concuerdan. Windows informa que hay un error en la configuración, sin embargo, no informa claramente que hay un conflicto entre las configuraciones.

Para resolver este tipo de problemas se requiere tener el disco del fabricante. En este disco esta contenido un programa de configuración. El programa de configuración pide los valores para cada una de las variables necesarias para la configuración del adaptador de red.

Los parámetros son comunes a muchos dispositivos como hemos visto en el capitulo anterior. Los parámetros IRQ, dirección de memoria base, dirección de puerto de I/O y acceso directo a memoria son utilizados en su mayoría por las tarjetas de red.

Aunque en la actualidad la mayoría de las tarjetas de red tienen la propiedad de ser Plug-and-Play, no todas lo tienen integrado. Es por esto que se requiere del disco de fabricante para poder configurar todas las opciones.

La mayoría de los valores que se requieren deben de ser los correctos, ya que de otra forma el adaptador de red no proporcionará el servicio. Para poder ajustar los valores correctos existen aplicaciones que permiten ver el estado del sistema y así poder decidir en que lugar de la memoria base o que interrupción puede o no la tarjeta de red.

Uno de los programas que se utilizaron para el diagnóstico de los valores disponibles es el checkit. Este programa muestra el mapa de memoria, lo cual es de gran ayuda para poder definir, en base al tamaño del programa controlador, la dirección de memoria base mas adecuada. Por lo regular, se asignan direcciones de memoria que van desde CC000 hasta DCFFF. Estos valores no son estándar, pero son los que mas se adecuan a las necesidades de las tarjetas de red.

Las interrupciones mas frecuentemente usadas para las tarjetas de red son la 7 y 5, sin embargo, si la computadora en la que se quiere instalar la tarjeta cuenta con un equipo multimedia, el valor deberá cambiar debido a que seguramente estas interrupciones han sido utilizadas para controlar los dispositivos multimedios.

Un valor que casi siempre es aceptado en la configuración de las tarjetas de red en la dirección de puerto de entrada/salida es el 280. Muy pocas veces se tienen problemas con este tipo de valores, sin embargo hay que revisar el equipo antes para cerciorarse de que no existirán problemas de conflictos entre puertos. El valor de acceso directo a memoria por lo regular se cambia a 2 o a 5, en este valor no hay tampoco mucho problema.

Otro programa para observar los valores internos de la computadora es el Microsoft Diagnostic (MSD). Este programa también realiza el diagnóstico del sistema de manera eficiente, sin embargo, el programa checkit realiza pruebas a la memoria, a los discos, etc. Gracias a esta herramienta podemos determinar con más claridad el origen del problema.

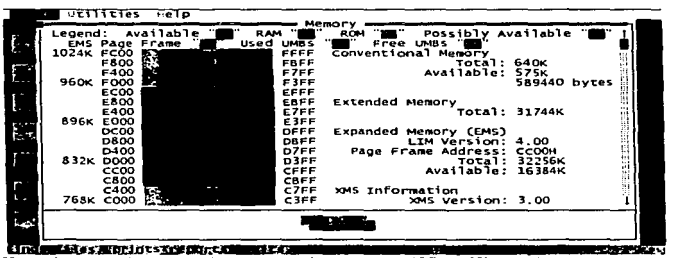


Figura 5.4 Mapa de memoria proporcionada por el programa MSD de Micrsoft

El valor de este tipo de aplicaciones no se ve sino hasta que se tienen problemas de conflictos de interrupciones o conflictos con las direcciones de memoria base.

Una vez que se tienen las herramientas necesarias para resolver problemas de configuración, se procede a implantar las soluciones.

Retomando el caso del laboratorio de cómputo, una vez que se detecto que el problema se debía a la incompatibilidad entre los parámetros físicos y los definidos por software, el siguiente paso fue resolver esta incompatibilidad.

Las tarjetas que se utilizaron estaban acompañadas de un disco de configuración, gracias al cual, se lograron configurar todas las máquinas de este laboratorio.

En la figura 5.5 se muestra la ubicación del laboratorio de cómputo dentro del edificio C.

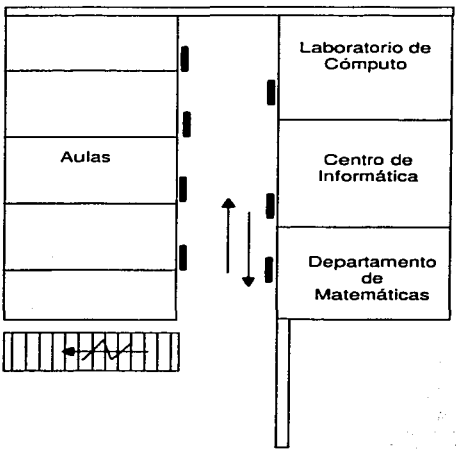


Figura 5.5. Edificio C

Dentro de este mismo laboratorio se realizó la implantación de TCP/IP en dos máquinas con Windows 3.11. La peculiaridad de estás máquinas era que, todos los parámetros de configuración de la tarjeta de red eran los indicados, sin embargo, iniciar Windows, la aplicación se congelaba.

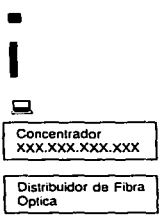
Al realizar el chequeo de todos los parámetros no se encontró ningún error. Se verifico que el controlador de la tarjeta no tuviera conflictos de ningún tipo con los programas de Windows, pero no se podría llegar a ninguna solución que diera resultados.

AL utilizar la herramienta de checkit, específicamente la parte de prueba de direcciones de memoria e integridad de la misma, se pudo llegar a la conclusión de que físicamente estaban dañadas algunas zonas de la memoria. Al correr la prueba, la aplicación daba resultados indeseables tales como el reinicio inesperado de la computadora.

Al cambiar los SIMMS de memoria por otros en buenas condiciones, la computadora fue capaz de iniciar los servicios de red sin problemas.

En el Edificio de Ingeniería Química se configuraron al menos 4 servicios de red. Este edificio presentaba problemas con la red. Estos problemas se debían principalmente a la alta cantidad de ruido, generado por la gran cantidad de equipo industrial. En general, la red de este edificio trabaja casi normalmente, ya que en ocasiones la red no soporta gran cantidad de ruido y pierde gran cantidad de paquetes.

La figura 5.7 muestra la forma en que están distribuidos los puntos de red en el edificio de Ingeniería Química. La simbología utilizada en este mapa y en aquellos mapas detallados se muestra en la figura 5.6.



Nodo de Red

Puerta

Computadora

Concentrador con Dirección IP asignada

Distribuidor de F.O

Figura 5.6 Simbología utilizada en los mapas de red.

En el transcurso de este capítulo se utilizarán dos tipos de mapas: mapa detallado y mapa genérico. El mapa genérico tiene como función ubicar el departamento en cuestión en el contexto del edificio o de una zona en específico. El mapa detallado muestra la ubicación de los puntos de red y de las computadoras en alguno de los departamentos.

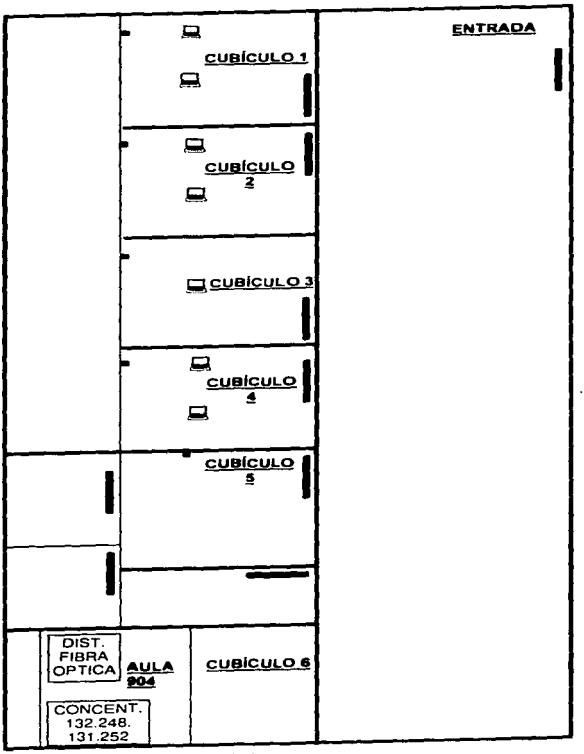


Figura 5.7 Laboratorio de Ingeniería Química.

El Centro de Informática es el alma de la administración de la red. En este departamento se concentra el personal que atiende todas las solicitudes de servicio a la red. Este departamento cuenta con la infraestructura necesaria para dar mantenimiento a los 4 segmentos de red. Sin embargo, su función no se detiene ahí, también se encargan de la programación y puesta en marcha del sistema de control de inscripciones.

El sistema de control de inscripciones anteriormente estaba implantado en un equipo HP. En la actualidad se ha realizado la migración de la base de datos a SYBASE en una computadora Sun Sparc. El sistema en general consta de dos partes, una que es el servidor de base de datos Sybase y otra que es el cliente Visual Basic.

Las computadoras cliente utilizan el software de Lan Work Place como soporte de red al cliente Visual Basic. En la puesta en marcha del sistema de inscripciones se presentaron algunas dificultades causadas por la alta demanda que se originó.

El problema que se presentaba era que, por lo regular se lograban establecer 5 sesiones, pero al intentar establecer una nueva conexión, cualquier otra de las máquinas perdía la conexión.

En un principio se pensó que el problema provenía del servidor. Se pensó que el servidor Sybase no tenia la configuración adecuada para dar soporte a mas de 15 procesos. Sin embargo, al verificar la configuración se comprobó que el servidor tenía mas capacidad de la exigida.

Dado que el proceso de inscripción es prioritario, la solución que se implementó fue desconectar temporalmente la mayor parte de los nodos del segmento. Esta solución aunque drástica permitió que el sistema de inscripciones funcionará correctamente. La solución podría haber causado conflictos con la mayor parte de los usuarios de la red, sin embargo, dado que el semestre aún no daba inicio y las labores bajaban su ritmo, la solución no causó demasiadas molestias entre los usuarios del segmento 131.

Las computadoras de desarrollo del sistema necesitan estar conectadas en red necesariamente para poder realizar el enlace con la base de datos. El sistema de red del centro de informática es bastante confiable. La red cuenta con un servidor de impresión, el cual esta implantado sobre Windows 3.11. La versión actual del sistema de inscripciones solo corre en Windows 3.1. Esto se debe principalmente a que la versión de Lan Work Place requiere de esta versión de Windows para poder funcionar. Sine embargo, Lan Work Place puede ser implantado sobre sistemas Windows 3.11 agregando un parche al sistema.

Algunas de las máquinas tienen un sistema dual de Windows, es decir, tienen una versión 3.1 para el desarrollo de las aplicaciones del sistema de Inscripciones y una versión 3.11 para el trabajo común en red. Aunque se realizó la propuesta de realizar la unificación de los dos servicios sobre el ambiente de Windows 3.11, el personal prefirió mantener el esquema utilizado ya que esto permitía estabilidad y consistencia en las aplicaciones desarrolladas.

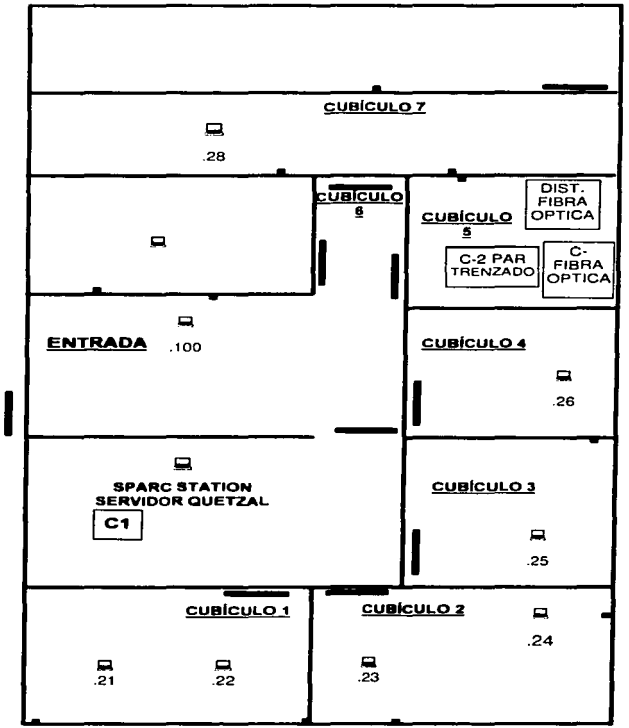


Figura 5.8 Centro de Informática

Aunque la red del Centro de Informática es bastante confiable, algunas veces presentaba problemas, especialmente con el servidor de impresión. Las dificultades mas frecuentes eran la falta de espacio en disco, aparentemente no debía de afectar el funcionamiento del desempeño de las labores de impresión, sin embargo, dadas las características de Windows 3.11, antes de realizar la impresión del archivo, realiza un spool en el disco duro local. Al no haber espacio suficiente en el disco, el servidor de impresión enviaba un mensaje de error inesperado en la conexión. Este problema era fácil de solucionar, sólo era necesario depurar el disco.

Otro de los departamentos en los que se brindaron varios servicios de conexión con TCP/IP fue el departamento de Matemáticas. En este departamento todas las computadoras utilizan los servicios de Red de Windows for WorkGroups. Aunque utilizaban los servicios de red de WfW, las computadoras no disponían de un recurso de impresión compartido. La mayoría de los usuarios debían imprimir sus documentos en forma directa, conectando una impresora al puerto de impresión.

Para dar de alta un servicio de impresión por red es indíspensable tener los controladores específicos para la impresora de que se trate. No importa si la computadora no este conectada directamente a la impresora, aunque sea una impresora remota, es necesario contar con los controladores.

La red de este departamento presenta un buen desempeño, las ocasiones en que se presentaron algunos problemas, se debió a la adición de un nuevo concentrador que fuera capaz de surtir los servicios de red para las aulas de clase y los auditorios.

La idea de agregar puertos de conexión en las aulas es una buena idea, ya que en cualquier momento que se requiera se puede tener acceso a la red.

Uno de los problemas que se enfrentaron en el departamento de matemáticas fue el hecho de que todas las computadoras tenían acceso a la red, pero sólo 1 no tenía el acceso a la red.

Lo primero que se revisa es la configuración de software. Se determina mediante solicitudes Ping si todo el stack de protocolo esta listo para ser utilizado. El ping despliega la información referente al estado del enlace entre la maquina local y la computadora destino, muestra información del numero de paquetes recibidos y el tiempo que se tardo en esperar los mensajes. Para obtener la información relevante al mismo host, se utiliza la dirección de loopback, es decir la dirección 127.0.0.1. Si el comando del ping devuelve los valores diferentes a request time out, quiere decir que todo el software instalado es capas de desempeñar las funciones de red.

La figura 5.9 muestra la configuración física del departamento de matemáticas. El servidor de impresión se encuentra ubicado en el cubículo de la secretaria. El servidor Sun Sparc Station no se encuentra funcionando, esto se debe a la falta de personal que se encargue de operar y de realizar la administración de dicho servidor.

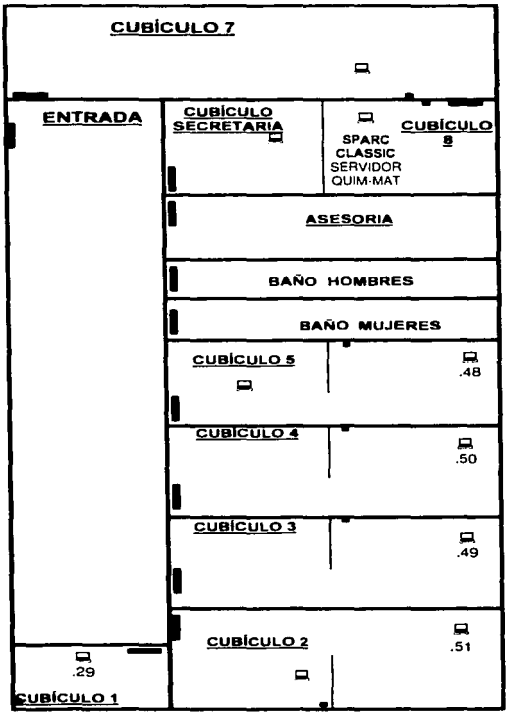


Figura 5.9 Mapa del departamento de Matemáticas.

En la Secretaria de Asuntos Escolares se realizó la instalación de Lan Work Place junto con los programas de aplicación para el sistema de inscripciones. En este lugar es donde se llevó a cabo toda la tarea de registro de alumnos en las diferentes materias. Los problemas de instalación se debieron principalmente a la falta de recursos de cómputo.

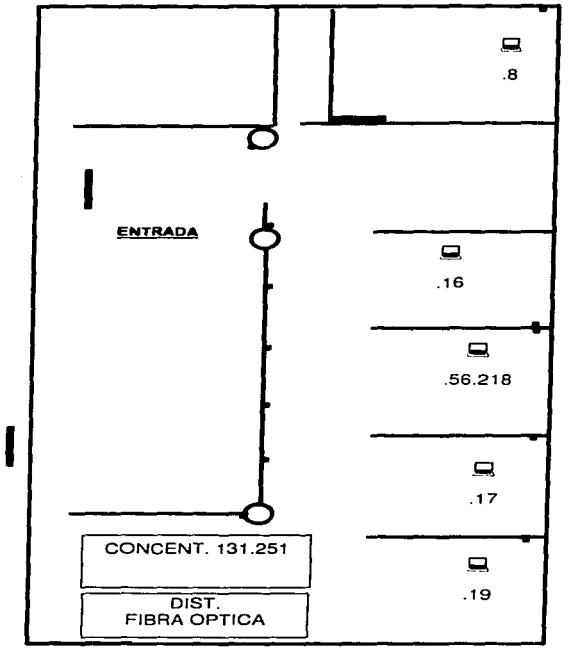


Figura 5.10 Departamento de Asuntos escolares

Los servicios brindados a otros departamentos de la Facultad son similares a los expuestos anteriormente. En el departamento de Coordinación de Carreras ser realizó una ampliación del segmento de red. Se colocó un concentrador conectado por medio de cable coaxial delgado. El concentrador realizaba su tarea perfectamente bien, sin embargo, no era capaz de conectarse con el concentrador en el lado opuesto. Se comenzó a investigar la causa probable de falla y se detecto que el puerto coaxial del otro puerto estaba dañado. Después de remplazar el puerto coaxial, la conexión se realizó con éxito.

En la Coordinación de Carreras se realizó la conectorización de todos los puntos.

En el área de Atención a alumnos también se realizo la instalación de un concentrador de 8 puertos, sin embargo, este fue conectado por medio de un cable par trenzado.

En toda la Facultad de Química se utiliza el estándar de cableado par trenzado EIA/TIA 568b. Este establece el siguiente orden para la conectorización:

Cable	Blanco/ Naranja	Naranja/ Blanco	Blanco/ Verde	Verde/ Blanco	Blanco/ Azul	Azul/ Blanco	Blanco/ Café	Café/ Blanco
Numero de Pin	1	2	3	6	4	5	7	8

Establecer estándares de conectorización permite ubicar de manera mas rápida los errores por falla en la conectorización. El Centro de Informática cuenta con un dispositivo cablerímetro que permite realizar pruebas en el cableado.

Para la realización de los mapas específicos de la red de la Facultad, el apoyo de esta herramienta fue muy valiosa, ya que permitió identificar la ubicación exacta de los puntos de red. Uno de los aspectos más importantes en que ayudó el cablerímetro fue la detección de errores. Una de las cualidades del cablerímetro mas notables es el mapa de cables. El mapa de cables permite ubicar la forma en que están conectados los cables en ambos extremos. En ocasiones es difícil distinguir en un cable ya conectorizado los colores, especialmente los cables en donde predomina el color blanco.

Tener un esquema general de red estándar permite que la respuesta a posibles fallas sea más rápida y eficiente. Si se tiene una red de TCP/IP bien administrada pero el cableado no tiene uniformidad, cualquier problema que surja podrá afectar a todos los niveles.

De igual forma, si se tiene una red con múltiples protocolos de forma desorganizada es muy probable que sea difícil encontrar los problemas en forma rápida, y más aún, encontrar una solución rápida que no repercuta en otras áreas.

Una de las partes más importantes de una red académica es la red de la biblioteca. En el edificio A se encuentra una biblioteca, la cual cuenta con una infraestructura de red bastante completa. Es a partir de ahí de donde se proporciona el servicio de red al área de atención a alumnos y la dirección.

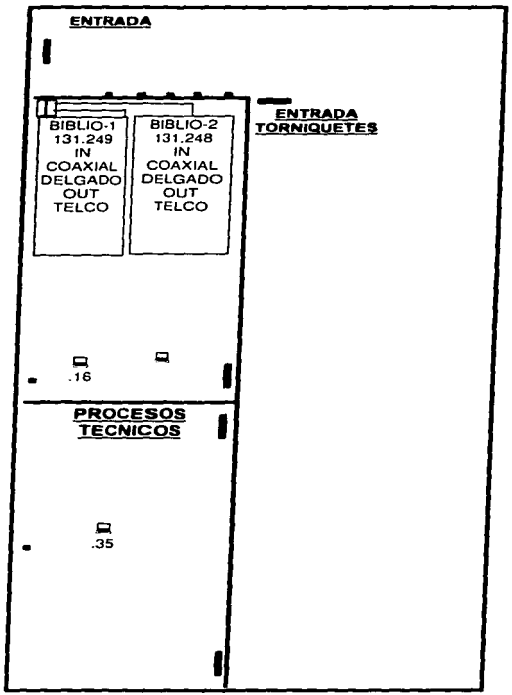


Figura 5.11. Red de la Biblioteca del Edificio A

La red de la biblioteca esta integrada por dos concentradores de 12 puertos cada uno, tal como lo muestra la figura 5.11

En la red de la biblioteca, los servicios están distribuidos por medio de un conector tipo telco, este tipo de conectores permiten tener 12 servicios de red en un solo cable. Esto permite centralizar aún mas la administración de la red. Sin embargo, como se podrá notar, se debe seguir un esquema para la separación de cada uno de los servicios, esto se hace por medio de un dispositivo conocido como regleta. La regleta esta formada por conectores plásticos con cuchillas que permiten realizar la conexión física entre el cable que proviene del conector telco y la salida a cada uno de los servicios individuales.

Hasta el momento sólo he descrito las actividades realizadas en el segmento 131. A continuación continuaré con los sucesos más importantes en la instalación de los servicios de red en el segmento 56, el cual está físicamente distribuido en todo el edificio B.

5.2. Implantación del segmento 56.

El segmento 56 es uno de los más complejos dentro de la estructura de red, ya que aunque es sólo un edificio, la arquitectura misma de este presenta serias dificultades para la ubicación de los departamentos que lo integran.

El edificio consta de 5 pisos, un sótano, una azotea y un anexo de posgrado. La red en la planta baja está integrada de la siguiente forma:

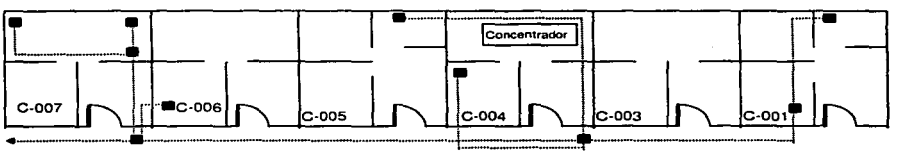


Figura 5.12 Planta Baja del Edificio B

La planta baja del edificio es una de las partes mas sencillas de la red de este edificio. Consta sólo de un concentrador de 12 puertos. Una de las particularidades de la planta baja es que cuentan con equipo macintosh, dada la poca frecuencia con que se utilizan estas computadoras dentro de la UNAM, integrarlas al ambiente de red no es una tarea sencilla. Sin embargo, una de las características del servicio de mantenimiento de la red que proporciona el Centro de Informática es resolver los problemas sin importar el tiempo que se deba invertir en ello.

Esta fue una de las primeras experiencias que tuve en la configuración de los servicios de red en ambientes macintosh. Está computadora pertenece al cubículo de Comunicación e Información de la Facultad. Su utilización es por lo regular para el diseño gráfico.

La mayoría de las implantaciones realizadas en los equipos del edificio B son comunes a las que he presentado anteriormente para el segmento 131. Sin embargo, me gustaría hacer mención de una computadora ubicada en el sótano del edificio, específicamente en el departamento de Rayos X.

La computadora de este departamento tiene una procesador 486 a 66 MHz, 8 Mb en RAM y un disco de 450 Mb. En un principio, el personal del departamento pidió ayuda para la instalación de un disco duro de 1 Gb. El sistema operativo en el disco duro original es Windows 95. Sin embargo, al intentar instalar el nuevo disco, el BIOS de la tarjeta madre no permitía el acceso a discos mayores a 500 Mb. Por esta razón se tuvo que instalar un programa que permitiera el acceso al disco. Desde el modo DOS el sistema operativo era capaz de reconocer el disco duro de 1 Gb, sin embargo, dentro del ambiente gráfico de Windows no era reconocido. Este problema persistió debido a que no se contaban con los discos originales de Windows 95. Posteriormente el personal realizó la petición de instalación del equipo multimedia y de los servicios de red. Esta computadora también cuenta con unidades de 5 ¼ y de 3 ½. Si hacemos la cuenta de cuantos dispositivos tenía que soportar la máquina nos daremos cuenta de que el número rebasaba el número de conexiones para alimentación de energía. Lo que se tuvo que hacer fue conectar los cables del drive de 3 ½ y de 5 ¼ forma compartida.

Para instalar los servicios de red se tuvieron varias dificultades, la primera de ellas fue la falta de IRQ disponibles. Este problema se solucionó haciendo combinaciones de configuración entre los dispositivos. Un vez solucionado este problema, se tuvo que configurar el numero de puerto de entrada y salida. Este problema fue fácil de solucionar debido a que la computadora permite aproximadamente 256 números posibles. Pero el problema que llevo mas tiempo corregir fue la dirección de memoria base.

El problema de asignación de memoria base implicó el uso de optimizadores de memoria. El problema se acrecentó debido a que tenían un emulador de Pentium, el cual tomaba una parte de la memoria para su funcionamiento. El programa para identificación del disco duro ocupa de igual forma una sección de la memoria. El controlador de la tarjeta de sonido y el del CD-ROM también ocupan un espacio en memoria. Por todas estas razones se tuvo que instalar el optimizador, ya que de otra forma habría sido una tarea titánica tratar de acomodar cada uno de estos controladores y programas en un área de memoria tan reducida.

También debido a que no se contaba con el software de Windows 95 original, la instalación de los servicios de red fue una tarea ardua. Se tuvieron que conseguir uno a uno los controladores idóneos para la tarjeta de red, ya que esta no contaba con los controladores para Windows 95.

Sin embargo, esta computadora fue una de las que mas conocimientos me aportaron, ya que fue todo un reto hacer que todos los dispositivos funcionarán. Aunque se contaba con una máquina más poderosa, una Aptiva, esta estaba dedicada solamente a la administración y utilización de un analizador de espectros.

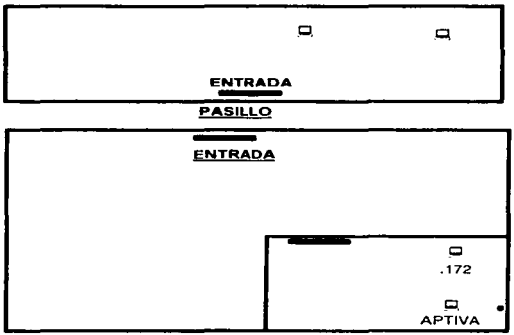


Figura 5.13 Red del departamento de Rayos X

Otro hecho que requiere especial mención fue la instalación de una torre de CD's, los cuales son utilizados para la consulta de bases de datos y de información en general.

Las torres de CD's se instalaron en el ambiente Windows para trabajo en Grupo. Se utilizó este ambiente porque es el que menor carga de trabajo tiene para la computadora y por esto, el acceso a los datos puede hacerse en forma eficiente.

En general, estos son los problemas a los que me he enfrentado durante la instalación de los servicios de red en la Facultad de Química, a continuación presentaré los mapas más relevantes dentro de la estructura de red de la Facultad.

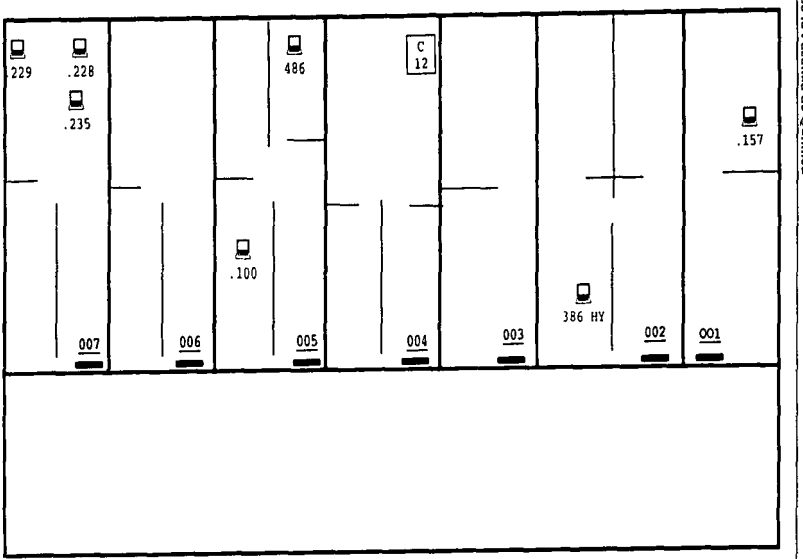


Figura 5.13 FUNAM

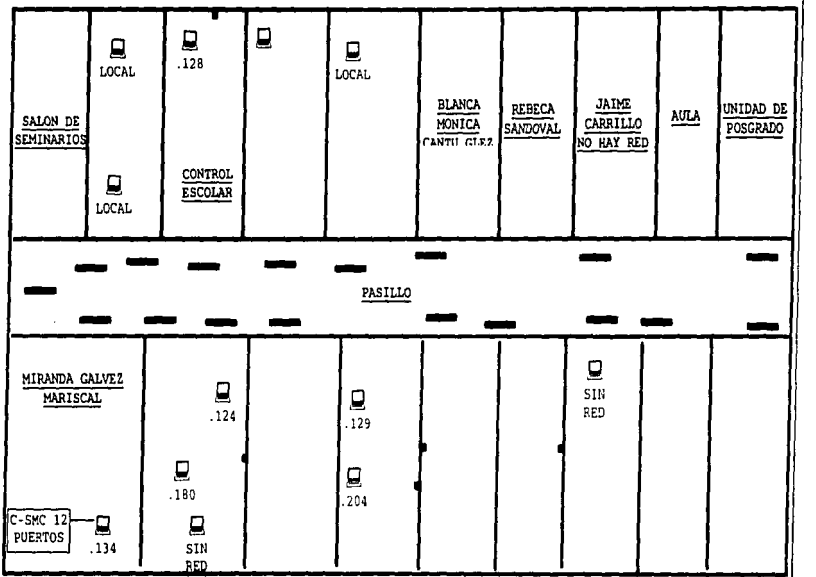


Figura 5.14 JEFATURA DE POSGRADO

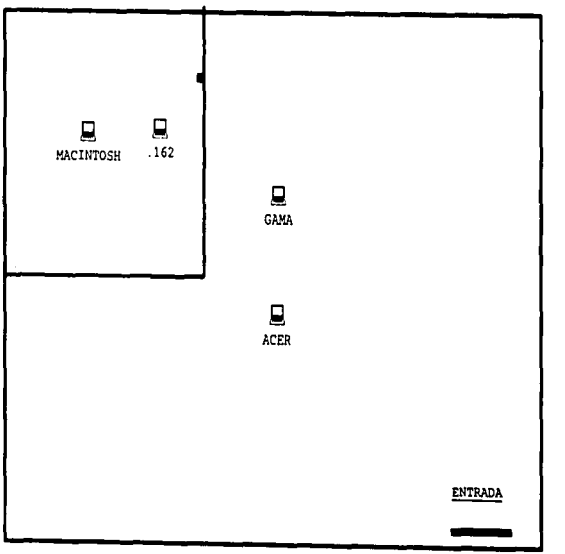


Figura 5.15 Laboratorio 101 Cromalografía de Gases

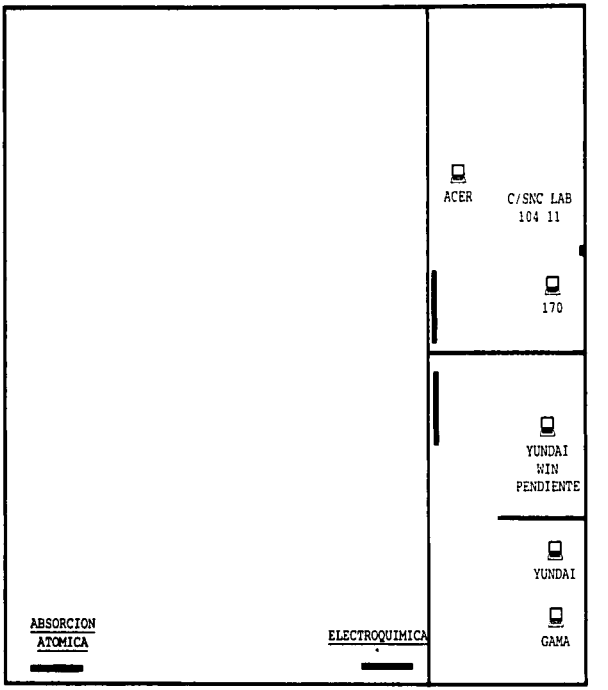


Figura 5.16 LABORATORIO 103

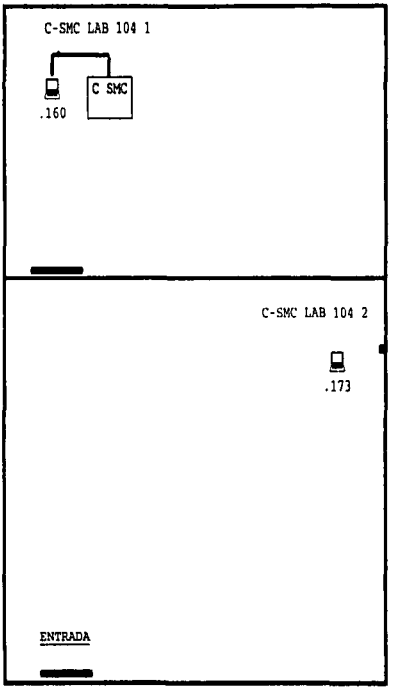
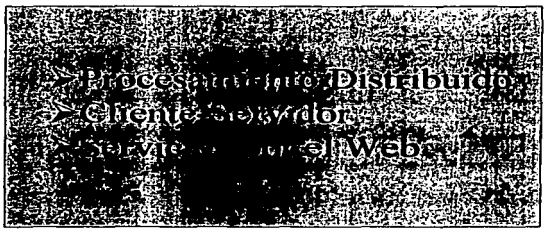
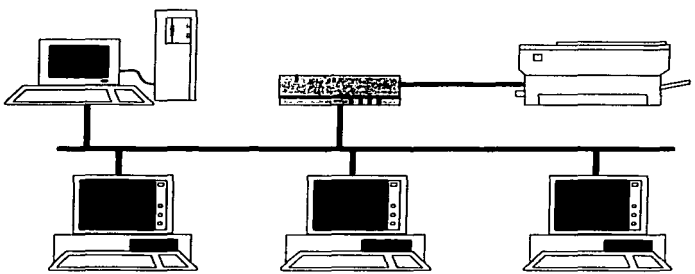


Figura 5.17 LAB 104 - ELECTROFORESIS



6

Explotación de los recursos de Red



Lista de Figuras

En este capitulo hablaré de la forma en que se pueden explotar los recursos de red. Al definir a TCP e IP, se están dando las pautas para el posible desarrollo de nuevas aplicaciones que realicen el trabajo de forma mas adecuada explotando todas las capacidades de TCP/IP. En la actualidad hay diferentes formas de explotar en serio los recursos de las redes de comunicaciones. Uno de esos esquemas es el modelo cliente-servidor.

6.1. El modelo Cliente-Servidor

El término servidor se aplica a cualquier programa que ofrece un servicio que se puede obtener en una red. Un servidor acepta la petición desde la red, realiza el servicio y devuelve el resultado de la petición. En el caso de los servicios sencillos, cada petición llega en un solo datagrama IP y el servidor devuelve una repuesta en otro datagrama. Un programa ejecutable se convierte en un cliente cuando manda una petición a un servidor y espera una respuesta. Debido a que el modelo cliente-servidor es de extensión conveniente y natural en la comunicación de interproceso una misma máquina, es fácil construir programas que utilicen el modelo para interactuar.

Los servidores pueden ejecutar tareas simples o complejas. Tareas tan sencillas como proporcionar el nombre de algún usuario o el nombre de la máquina. Los servidores se suelen implantar como aplicaciones de programas. La ventaja de implantar los servidores como programas de aplicación es que pueden ejecutarse en cualquier sistema computaciona que soporte la comunicación TCP/IP. De este modo, el servidor de un servicio en particular puede ejecutarse en un sistema de tiempo compartido junto con otros programas o en particular pueden ejecutarse en la misma máquina o en múltiples maquinas. De hecho, los administradores comúnmente duplican copias de un servidor dado en máquinas físicamente independientes para incrementar la disponibilidad o mejorar la ejecución.

La manera más simple de interacción cliente-servidor se vale del envío de un datagrama no confiable para transportar mensajes de un cliente a un servidor y de regreso. Consideremos por ejemplo, el servidor de eco de UDP. En el lugar del servidor se inicia un proceso servidor de eco UDP. Para poder llevar a cabo esto, el servidor debe negociar el permiso para poder iniciar el servicio y obtener el puerto conocido . Una vez que se ha obtenido el permiso, el servidor de eco se queda en espera de algún datagrama que solicite sus servicios. Cuando llega un datagrama, primero se debe examinar el datagrama y procesar el datagrama para extraer las direcciones de origen y destino. Al tener las direcciones origen y destino, estas se invierten y el datagrama se envía de regreso al originario del mensaje.

Del lado del cliente, primero se realiza la solicitud de eco hacia algún servidor y se espera la respuesta. En este ejemplo se muestran detalles esenciales del modelo cliente-servidor. El primero de ellos tiene que ver con los tiempos de vida de cada uno de los participantes de la comunicación.

Los servidores por lo regular tiene un tiempo de vida mas largo, ya que tienen que estar atentos a todas las peticiones que realizan los clientes. Por el contrario, los clientes sólo se mantienen ejecutando mientras realizan la petición y esperan los datos.

Otro aspecto que se presenta en este ejemplo es la asignación de puertos. Como ya he dicho antes, hay aplicaciones que tienen puertos bien conocidos, como telnet, ftp, etc. Los servidores necesitan tener asignado un puerto al cual todos los clientes puedan realizar sus peticiones. Podemos realizar una analogía con las tiendas. Las personas que quieren vender sus productos o servicios anuncian su ubicación y los clientes van al lugar a realizar las compras. En el modelo cliente-servidor sucede algo similar, los servidores tienen un número de puerto fijo y los clientes pueden tomar el que deseen.

El ejemplo anterior es bastante simple ya que las peticiones son secuenciales, es decir, se procesa una petición a la vez. Después de aceptar una petición, el servidor forma una respuesta y la manda antes de volver a ver si ha llegado otra petición. El servidor asume que el sistema operativo hará una cola de espera para los datagramas que lleguen antes de poder procesarlos.

En la práctica los servidores suelen ser mas difíciles de construir que los clientes, ya que necesitan acomodar varias peticiones concurrentes, aun cuando una sola petición se lleve una cantidad de tiempo para ser procesada. Por ejemplo, consideremos que un servidor de transferencia de archivos es el responsable de copiar un archivo a otra máquina bajo petición. En general, los servidores tienen dos partes. Un programa maestro sencillo, responsable de aceptar nuevas peticiones y un conjunto de esclavos, los responsables de manejar las peticiones individuales. El servidor se encarga de abrir el puerto, esperar el cliente, iniciar el esclavo que procese la tarea y volver al estado de espera.

Como el maestro inicia un esclavo para cada nueva petición, el procesamiento procede de manera concurrente. De este modo, las peticiones requieren poco tiempo para completarse se pueden terminar antes de que las que toman mas tiempo independientemente de la que se haya iniciado primero.

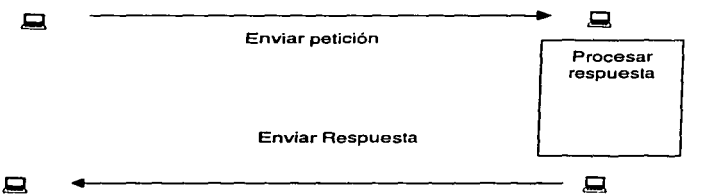


Figura 6.1. Modelo de la tecnologia cliente-servidor

Además de la complejidad que resulta de que los servidores manejen peticiones concurrentes, la complejidad también surge porque los servidores deben reforzar las reglas de autorización y protección. Los programas servidor suelen requerir una ejecución de alta

prioridad pues tienen que leer archivos del sistema, mantenerse en línea y tener acceso a datos protegidos. El sistema operativo no restringirá un programa servidor si intenta tener acceso a los archivos del usuario. De este modo, los servidores no pueden cumplir a ciegas las peticiones de otras localidades. Por el contrario, cada servidor toma la responsabilidad para reforzar el acceso al sistema y las políticas de protección.

Por último, los servidores deben protegerse a si mismos contra las peticiones formadas equivocadamente o contra las peticiones que causarán que el mismo programa servidor ser aborte. A menudo es difícil prever los problemas potenciales.

En el contexto de los sistemas de información, el término cliente-servidor toma un significado diferente en algunos aspectos. En los sistemas de información se tiene un programa encargado de proporcionar una vista agradable a los datos y otro programa encargado de enviar los datos que se van requiriendo, al primero se le conoce como cliente y al segundo servidor. También se les conoce como front-end y back-end.

En un esquema de tipo cliente-servidor el servidor no realiza todo el procesamiento, ahora el cliente sólo va a requerir el envío de datos, quizá hasta un poco de procesamiento, pero en general, el cliente realizará peticiones de datos en crudo. El cliente mismo se encargará de realizar el procesamiento de los datos. La tarea del servidor consistirá en atender las peticiones de datos y enviarlas directamente al servidor. El cliente realizará las operaciones necesarias para mostrar los datos de forma amigable y sencilla para el usuario.

El ejemplo más sencillo de un ambiente cliente- servidor son los browsers para Internet, tales como el Netscape Navigator, el Microsoft Intenet Explorer, Mosaic y otros. En este caso, el servidor puede ser cualquier programa servidor de páginas, tales como Apache, http de la NCSA y otros. En este ambiente cliente-servidor, el cliente hace las peticiones necesarias para obtener las páginas en html. Una vez que ha recogido toda la información, la procesa para poder realizar el despliegue de la página en sí.

En el ejemplo anterior realmente se está traduciendo código, ya que html define un estándar para la comunicación entre diferentes equipos. El definir estándares de html ha permitido un gran auge de las tecnologías de información vía internet.

6.2. Sistemas distribuidos.

En un principio, los sistemas se conectaban por medio de cables a unidades centrales de procesamiento, toda la programación se realizaba por medio del intercambio de cables. Posteriormente, esta labor fue realizada en tarjetas perforadas, las cuales tenían el inconveniente de si se llegaba a perder alguna de ellas, todo el programa se perdía. Posteriormente aparecieron las primeras PC's, las cuales impulsaron en gran medida el desarrollo de la computación. La computadora dejo de ser un Mito para convertirse en una herramienta más en la oficina y el hogar. A partir de la aparición de las primeras PC's, comenzó a descentralizarse el poder de cómputo, utilizandose cada vez más las redes de área local.

La tecnología siguio su paso ascendente y en cada nueva versión de sistemas PC el procesador se iba haciendo más robusto. Esto permitió que las computadoras personales alcanzaran niveles de procesamiento bastante altos. Por ejemplo, las computadoras

personales hoy en dia pueden hacer las mismas tareas que las computadoras de hace 20 años. Sin embargo, hay muy pocas aplicaciones realmente distribuidas. La mayor parte de las aplicaciones que utilizamos hoy en día son productos de hace bastantes años, por ejemplo telnet y ftp.

El desarrollo de los sistemas distribuidos puede ser visto en términos de las arquitecturas utilizadas para describirlos. En 1985, Tanembaum describió tres diferentes formas de arquitecturas de sistemas distribuidos:

1. Modelo de minicomputadora: En este modelo, hay un pequeño número de minicomputadoras, cada uno con varios usuarios, que están interconectadas por un medio de un switch crossbar.

2 Modelo de pool de procesadores. En este modelo un procesador es asignado con base a la necesidad de un pool que se encuentra en su lugar. No había el concepto, de propiedad de máquina uno carga al sistema y se le asignaba una máquina mientras el usuario esta dentro del sistema. Solo un usuario podría estar dentro del de una máquina en todo momento.

Modelo WS

Cada usuario tiene una WS que tiene un procesador poderoso. Existe acceso a servidores de archivos e impresión de red. Los primeros dos modelos parecen ya pasados de moda. Ha habido investigación reciente respecto a lo que se conoce como red de computadoras de escritorio, que consiste en un numero de elementos computacionales conectados pero no por un bus sino por una red ATM. Esta red para opera obviamente mediante un switch de ATM, haciendo ver a todo el sistema como el modelo minicomputadora. Además, la segunda forma de sistema podría ser parecida a una serie de terminales X que contengan un poco mas de poder de que es necesario para soportar la interfaz gráfica y que son actualmente estoy por comenzar una nueva vida en el área de sistemas

Modelo integrado. El último modelo, que parece el actual, esta en cierto modo alejado de la dirección de la investicación de sistemas distribuidos.

Sin embargo, hasta ahora he hablado de sistemas distribuidos, pero que son en realidad los sistemas distribuios? La respuesta depende de quien se realice la pregunta. Sin embargo, en el cómputo paralelo se pueden clasificar ciertas clases de máquinas paralelas como distribuidas. Por ejemplo:

SIMD (Single Instruction Stream, Multiple Data Stream computers) son aquellas que tiene un número de elemtentos de procesamiento queejcutan las mismas instrucciones por pasos bloqueados, pero en diferentes bits de información. Son buenas para aplicaciones que tienen un alto grado de regularidad y que por lo tanto se usaron para aplicaciones con arreglos o vectores. Son extremadamente mas caras..

MIMD (Multiple Istruction stream Muiltiple Data strams computers) son la forma mas común de multiprocesadores y de memoria compartida. Tiene series de procesadores de bajo costo cada uno con su propio cache, pero con acceso a memoria común. Ejemplos el

trasputer. El objetivo de estos sistemas es ganar desempeño paralelizando los cálculo o ganar confibailidad por el medio de repolicación.

Al aumentar en número de procesadores (n) los costos de comunicación se elevan a 2n aproximandamente.

Construyendo sistemas SIMD o MIMD implica hechos que son de interes. No obstante, los sistemas debilmete acoplados que son sistemas que están formados por un número de nodos conectados por medio de una red, aun son las generales por el hecho de que hay problemas que surgen en ellos y que no aparecen en la bitacora.

Se asumirá lo sigueinte acerca de los sistemas debilmente acoplados y que vermos mas tarde):

Los nodos tienen modo independiente de falla. Es posible que uno de los nodos en el sistema falle nos va a des inafectar la operación del resto.

Los enlaces ligas o uniones también tienen modos independientes de la falla. Es posible que un enlace falle sin afectar la comunicación en otro lado o el procesamiento

Los nodos están normalmente distribuidos en área geográfica (hace hasta poco esto se referia a una red local, pero esto se ha convertido en redes de area amplia). Por lo que el tiempo de espera en la transmisión de datos entre ellos es algo si se compara el tiempo de proceso de los nodos.

Debido a que las máquinas son tipicamente estaciones de trabajo independientes, es razonable el asumir que habrá un número diferente de usuarios que posiblemente pertenezcan a organizaciones diferentes, que interactuan y posiblemente hostiles.

En general podemos asumir que hay una gran cantidad de heterogeneidad en el sistema: los procesadores pueden ser de diferentes tipos, pueden estar corriendo diferentes paquetes de software y quieren interactuar, pueden haber diferentes sistemas operativos, diferentes sistemas de archivos, convenciones de nombres, procesos administrativos, etc.

Hay un cambio constante dentro de un sistema de mayor escala: las maquinas siempre estan siendo apagadas, movidas, reconfiguradas, actualizadas, etc.

No todos los sistemas debilmente acoplados obedecen todas estas reglas. Claramente, es posible tener un sistema que consista solo de un numero similar de maquinas en un anillo local, no conectado a algo. Esto viola muchas de las cosas asumidas anteriormente: la falla de cualquiera de los nodos de cualquier enlace no afecta a otros, y que el sistema es de gran escala, heterogenero. Etc. Sin embargo, estas son los supuestos que podemos hacer y claro es que podemos hacer y es claro que podamos tratar con ellos en casos generales. Podemos al menos producir sistemas que funcuionen localmente.

6.2.1. Pros y contras de la Distribución

Por lo regular, el tipo de problema determina la solución. A continuación se muestran algunas ventajas de implantar sistemas distribuidos.

- Costo y extensibilidad. Hay un desembolso menor para un sistema distribuido con el mismo poder de procesamiento que un mainframe centralizado; los sistemas distribuidoras no necesitan plantas enfriadoras dedicadas, espacio dedicado, etc. Mas aún, cuando uno compra un mainframe, uno debe comprar mas de lo que es inicialmente requerido para el futuro, debido a que el uso tiende a incrementarse con el tiempo.

- Es fácil confeccionar un sistema distribuido a las necesidades del moneto. Por mucho, el extender el sistema distribuido es facil; uno solo enchufa una máquina mas a la red, cambia algunas tablas y listo. Con un mainframe , uno debe dejar "bajar" la maquina que todos los usuarios se salgan del sistema y efectuar la actualización en un tiempo no breve.

- Autonomia y usabilidad. Las mainframes no fueron diseñadas par procesamiento interactivo, son mucho mejore para procesamiento en batch. WS son mejores para implantar el tipo de interfaces gráficas altamente interactivas que los usuarios demandan en estos dias. También son más fáciles de configurar para el personal de lo que puede ser un mainframe. Aún mas, las estaciones de trabajo sin utilizar siguen siendo parte de la base de computadoras aún cuando no se está usando; sus ciclos pueden ser usados para ejecutar procesos para otros usuarios. Uno puede desconectar partes de un sistema distribuido que después podrá correr autónomamente. Un ejemplo podría ser la preparación de exámenes finales a los cuales estos no tienen acceso por ninguna vía.

- Compartición de recursos. En contraste con el deseo de la autonomía, si sitios en un sistema tienen diferentes capacidades, el cómputo distribuido permitirá una mayor utilización de los recursos disponibles. Por ejemplo, algunos nodos podrían tener dispositivos de almacenamiento o impresoras conectados y, algunos de ellos podrían tener hardware especializado como un arreglo de procesadores. Lo mismo puede ser cierto para recursos computacionales; en algunas aplicaciones son mejor en estructuradas como una serie de servicios especializados que pueden ser proporcionados cada uno de ellos por un servidor dedicado.

- Mayor velocidad de computo, Muchas aplicaciones que pueden estar estructuradas como una serie de sub-calculos. Si estas pueden ser ejecutadas en paralelo, posiblemente se tenga una ganancia en la velocidad de procesamiento. Aún más, puede ser posible migrar procesos a otros cuando el nodo en el que se encuentran con que esta sobrecargado para lograr mayor velocidad, Sin embargo, ninguno de estos hechos implica que podemos obtener beneficio total de este punto de vista,

En algunas aplicaciones demasiada comunicación y problemas de planeación del uso del procesador disminuyen las ventajas ganadas por la distribución. Es dificil para sistemas que balancean la carga determinar anticipadamente cual de los procesos que están siendo ejecutados en un nodo propiciaran mucha comunicación en el futuro. Como resultado no es generalmente posible hacer predicciones del efecto de mover un objeto.

- Confiabilidad. Si un Nodo falla, es posible que el resto de los nodos siga funcionando, aunque posiblemente exista una degradación en el desempeño. Es muy poco probable que un sistema de seguridad critico que toda la capacidad de procedimiento falle. Claramente si el sistema esta construido con un uniprocesador, esto es difícil de asegurar. Sin embargo, si la información relevada es replicada entre un grupo de nodos,

entonces el fallo del nodo probablemente no afecte el procesamiento y el sistema puede continuar operando. Además, la replicación permite mayor disponibilidad de servicios o de información dando mayores ventajas en términos de velocidad de procesamiento.

- Comunicación. Existen aplicaciones que son inherentemente distribuidas y para las cuales no existe razón de tenerlas en sistemas centralizados. Estas implican comunicación entre nodos y usuarios geográficamente separados. El ejemplo más obvio de esto es el correo electrónico.

Las desventajas de los sistemas distribuidos son debidas al hecho de que están geográficamente distribuidas y por lo tanto la seguridad de la información de las computadoras es mucho mas fácil de controlar y son considerablemente mas complejos que los sistemas centralizados, lo que significa que son más difíciles de administrar y menos predecibles en términos de comportamiento.

6.3. Servicios por el Web

El World Wide Web es una de las tecnologías más importantes de nuestros dias. Esta tecnología ha reunido a miles de máquinas alrededor del mundo. Su servicio se basa en html, un lenguaje de hipertexto. El hipertexto es muy similar a la ayuda que presenta Windows, es decir, permite ir saltando entre los términos mas importantes. En los servicios tradicionales de navegación, todo era texto, casi no habían gráficos.

Las primeras versiones de html ya permitian utilizar gráficos y texto. Sin embargo, los navegadores comenzaron a madurar y derivaron el los browsers que hoy conocemos: netscape, internet explorer y Mosaic.

Hoy en día los sistemas de información han encontrado en el World Wide Web una herramienta de desarrollo muy versátil. El WWW proporciona en forma nativa un cliente universal, ya que la mayoria de las veces no importa que software se utilice para navegar.

Los CGI's (Common Gateway Interface) proporcionan una interfaz entre el usuario que entra a una página y el servidor de http. Gracias a los CGI's podemos encontrar en la red páginas que solicitan datos. Estos datos son atrapados por programitas escritos en C, perl, shell o algún otro ejecutable. Estos programitas son los CGI's. Los CGI's permiten interactuar con las páginas de Web. Por ejemplo, se puede tener una forma de inscripción de alumnos en donde cada alumno puede introducir sus datos, estos datos los atrapa el CGI y los envía al servidor de Base de datos.

Los CGI's funcionaron de maravilla hasta la aparición de Java, este lenguaje programación está revolucionando el mundo de las redes.

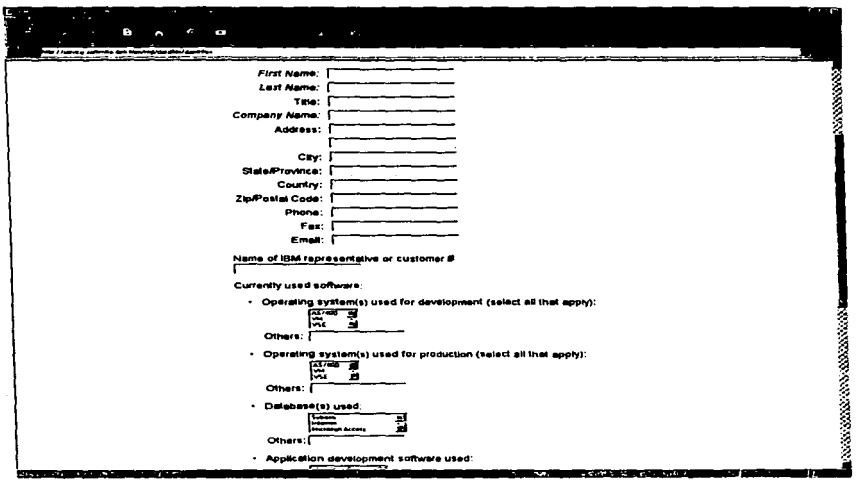


Figura 6.2 Ejemplo de una página html con espacios para llenar datos.

Esta página html es conocida como forma, los datos introducidos son atrapados por un CGI, el cual los procesa y si es necesario, vacía los datos en una base de datos.

Java es un lenguaje que tiene características muy especiales, ya que puede correr en casi cualquier plataforma sin necesidad de volver a compilar. Java proporciona librerías de conexión con TCP/IP. Con las librerías de TCP/IP podemos realizar la programación de casi cualquier cosa para la red.

Los applets son programas realizados en Java especiales para el Web. Con los applets se pueden realizar menús activos, similares al de cualquier aplicación, se pueden crear animaciones, programas para hablar con otras personas (chat), etc.

En la figura 6.3 se muestra el ejemplo de un applet para la creación de curvas por cuatro métodos distintos.

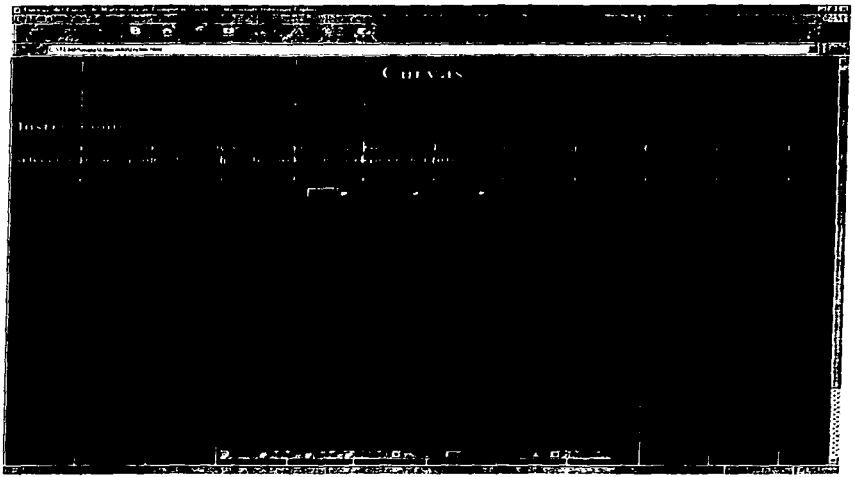


Figura 6.3. Ejemplo de un applet.

Otra de las opciones para la creación de páginas interactivas son los componentes Activex de Microsoft.

La desventaja de los Activex es que sólo pueden ser visualizados con Internet Explorer, Netscape (hasta la versión 4) no soporta este tipo de componentes. Sin embargo, los componentes Activex son bastante buenos, con ellos se pueden lograr aplicaciones para el Web similares a las que se podrían desarrollar en Visual Basic o Delphi.

Conclusiones

Los sistemas de información de la actualidad están siendo orientados al uso de la red como un recurso mas. TCP/IP es el protocolo ideal para el desarrollo de las nuevas tecnologías.

TCP/IP ha demostrado en la actualidad ser por mucho, la mejor opción para la interconexión de sistemas heterogéneos. En un principio TCP/IP sólo se utilizaba para sistemas de gran tamaño, en cambio hoy puede ser utilizado conectar los sistemas mas sencillos hasta los mas complejos.

La Facultad de Química es un claro ejemplo de la diversidad que puede haber en el campo de la computación, ya que es posible encontrar equipos que sería muy difícil de encontrar en otros lugares.

El desarrollo de estándares para la implantación de TCP/IP en equipos personales ha permitido un gran auge en el desarrollo de la tecnología de redes locales, las cuales han dado un gran impulso a la tecnología de información distribuida.

Internet se ha convertido en un gran gigante en donde conviven distintos tipos de computadoras y software. TCP/IP es la base para que todas las computadoras que integran Internet puedan comunicarse. Gracias a que es un protocolo adoptado por la mayoria de los desarrolladores de software, se han desarrollado aplicaciones que trabajan en diferentes sistemas operativos.

Gracias a protocolos como TCP/IP y lenguajes de programación como JAVA es que hoy podemos utilizar las mismas aplicaciones no importando el sistema operativo ni la computadora que se utilice.

Hoy en día estamos viviendo la revolución de la información, en donde aquellos que tengan los medios de comunicación mas eficientes serán los que destaquen y los que logren un desarrollo significativo. El México de hoy debe estar consiente de que las tecnologías de información distribuidas le permitirán alcanzar un desarrollo consistente. Sin embargo, no es necesario empezar de cero, ya que existen los medios necesarios para lograr un crecimiento informático.

TCP/IP se presenta como la alternativa más viable para el desarrollo de redes de alta confiabilidad y el desarrollo de aplicaciones que trabajen con la red como un recurso más.

Bibliografía

- **BLACK** Uyless
  Redes de Computadoras. Macrobit. 1987

  TCP/IP and Related Protocols. Mc Graw-Hill 1992

- **CHORAFAS** Dimitris N.
  Local Area Networ Reference. Mc Graw-Hill. 1989

- **COMER** Douglas E.
  Interworking with TCP/IP. 2da e.d. 1991

  Vol I. Principles, Protocols and arquitecture

  Vol. II Design, Implementation and Interals.

  Redes Globales de Información con Internet y TCP/IP. Principios básicos, protocolos y arquitectura

- **FEIT** Sidnie Ranade Jay
  TCP/IP Architecture, Protocols and Implementation. Series Advisor. McGraw-Hill. 1993

- **HUNT** Craig
  Networking Personal Computers with TCP/IP. O'Reilly & Associates. Inc. 1995

- **HSU** John Y.
  Computer Networks. Architecture, Protocols and Software. Arch House. 1996.

- **NUSSBAUMER** Henri
  Computer Communication System. Vol. 2. Wiley. 1990

- **SIYAN**, Karanjit S. **RUBACZYK** Peter. **KUO** Peter.
  Interworking with Netware TCP/IP. New Rides. 1996

- **TANENBAUM** Andrew. S.
  Computer Networks. 3raed. Prentice Hall 1996