75

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

SEGURIDAD EN REDES DE COMPUTADORAS

T E S I S

Que para obtener el título de

INGENIERO EN COMPUTACION

p r e s e n t a

SILVIA RAMIREZ GARCIA

DIRECTOR DE TESIS: ING. ANTONIO GARCES GARCIA
CODIRECTOR DE TESIS: M. I. ADOLFO MILLAN NAJERA

México, D. F. 2000

Agradecimientos

A mis padres Hermilo Ramírez y Silvia García
por su apoyo en todos los aspectos de mi vida.

A los maestros que me guiaron y transmitieron
sus conocimientos durante mi carrera personal.

A mis amigos que me brindaron su ayudado
cuando se las solicité.

Un libro abierto es...
un cerebro que habla;
cerrado...
un amigo que espera;
olvidado...
un alma que perdona;
destruido...
es un corazón que llora.

Índice

Introducción

La aparición de la computadora fue el resultado, de entre muchos otros problemas, de la necesidad de simplificar el proceso de cálculo número, sin embargo su aplicación ha ido más allá de esta tarea: procesos de control, simplificación de tareas rutinarias y hasta peligrosas, y en nuestro caso aplicaciones en el área de las comunicaciones. La computadora tiene la capacidad de generar grandes volúmenes de datos que en la mayoría de los casos es necesario analizarlos en diferentes aspectos, para ello en ocasiones se requiere que este conjunto de datos sea transferido a personas y equipos en diferentes sitios; por lo cual se requiere que exista una interconexión de tal forma que pueda darse esa transferencia de datos en una manera rápida y eficiente. Esto genéricamente se denomina red de computadoras.

Hoy en día, la información se ha convertido en un sujeto de valor económico y estratégico para las personas, las empresas y países; y es posible que alguien más tenga interés en la información que se transfiere, también puede haber interés en que ésta no llegue a su destino o llegue alterada. Actualmente existen herramientas y recursos de cómputo que realizan estas tareas, y se han difundido de tal forma que su facilidad de obtención y de aplicación pone en constante riesgo la integridad y confidencialidad de la información.

El presente trabajo se centra en el tema de seguridad en redes de computadoras y sus principales objetivos son los siguientes:

- Diseñar e implementar una aplicación que realice transacciones de información segura entre los diferentes departamentos del Centro de Instrumentos de la UNAM. Estas transacciones involucran el envío de información confidencial que comúnmente se desarrollan en línea: envío de datos sobre el estado de ciertos recursos económicos; ideas sobre proyectos académicos o administrativos, es decir, se realizan transmisiones de mensajes con información que no debe ser accesada por cualquier miembro de la comunidad.
- Generar material estructurado, de tal forma que sirva como referencia en la materia de redes de computadoras.

El cuerpo de la tesis consiste de cinco capítulos. En el capítulo uno tratamos los antecedentes históricos y la evolución de las redes de computadoras, así como también el origen de la seguridad de las mismas y su estado actual. Así mismo, en el capítulo dos y tres se presentan los protocolos de comunicación con lo cual podemos encontrar sus características intrínsecas en cuanto a su seguridad, sin embargo, dado a que éstos son difícilmente alterables en su estructura, en el capítulo cuatro se analizan las posibilidades de fortaleza a nivel aplicación, la transferencia de información por medio de la criptografía y también se presentan algunas aplicaciones actuales en el ámbito comercial y gratuito.

En el capítulo cinco se hace una propuesta de solución al problema de transacciones seguras utilizando el conocimiento expuesto en los capítulos anteriores, así como de herramientas de programación.

Finalmente, se incluyen las conclusiones y un disco compacto que contiene cuatro apéndices, un glosario y la aplicación de seguridad EncripData, con el propósito de complementar la información necesaria para la realización de este trabajo.

Capítulo 1

Antecedentes Históricos

El origen de las redes de computadoras no surgió de forma espontánea, más bien fue el resultado de la unión de varias ideas e inventos que se dieron a través del tiempo. Su principal objetivo era y sigue siendo la transferencia de información en forma eficiente y "segura". Esto se ha ido logrando gracias al surgimiento de las diferentes tecnologías relacionadas con el manejo de datos. Para saber el por qué de estas tecnologías y del enorme crecimiento que han tenido las redes de computadoras debemos revisar primero su evolución.

1.1. Antecedentes históricos de las redes de computadoras.

Sin remontarnos a la creación de las primeras máquinas calculadoras, partiremos de un hecho muy importante en el manejo de información: la invención de las máquinas de tarjetas perforadas. La consolidación de esta idea la llevó a cabo Herman Hollerith cuando realizó el censo norteamericano de 1890. Él se dio cuenta que la mayoría de las respuestas eran afirmaciones y negaciones, que podían representarse como la presencia o ausencia de una perforación en una tarjeta. También en nuestro país, este tipo de registro fue implementado en Ferrocarriles Nacionales de México en el año de 1927, el cual, consistía en la introducción de datos en la tarjeta (perforación); su elaboración (clasificación, intercalación, cálculo, etc.); y la obtención de datos.

Por otra parte, en el área de las comunicaciones después de la aparición del telégrafo y del teléfono, surgió un nuevo sistema de comunicación que consistía en la instalación de radios emisores en las patrullas del Departamento de Policía de Detroit de los Estados Unidos. Este hecho fue la pauta para que en el año de 1947, AT&T Bell empezara a desarrollar el concepto celular.

Las necesidades de manejo de información se acrecentaban al igual que las investigaciones científicas en este campo dando como origen a la primera generación de computadoras. En 1946, se realizó el proyecto ***ENIAC*** que consistía en la creación de una máquina calculadora capaz de resolver los problemas de balística usando como tecnología los tubos de vacío. Sin embargo, en el año de 1952 empiezan a funcionar las máquinas "Von Neumann" (en honor a su inventor), cuya innovación fue el almacenamiento de programas. A partir de esta fecha se inició la etapa de los procesadores electrónicos y la sustitución de los tubos de vacío por transistores.

En la segunda generación de computadoras podemos mencionar la aparición de la computadora IBM701, la cual se caracterizó por su memoria basada en núcleos magnéticos. La ventaja de este tipo de memoria fue el registro de un número mayor de datos en un espacio menor y la lectura era miles de veces más rápida en comparación de la máquina de Von Neumann. También IBM se hizo presente en la Universidad Nacional Autónoma de México, al instalar el procesador IBM-650 con el propósito de realizar cálculos en la investigación científica y universitaria.

Finalmente, el impulso decisivo en el desarrollo de los procesadores electrónicos (construidos mediante chips) y particularmente en el proceso de datos a larga distancia lo determinó el lanzamiento del satélite Sputnik, construido por la Unión Soviética en 1957.

Después de este logro, Estados Unidos deseaba estar en la cabeza de la tecnología militar por lo que decidió reestructurar su red telefónica para poder contar con una red de comando y control que sobreviviera a cualquier guerra. Es por ello que en el año de 1969 surge ARPANET, una red que aseguraba la transmisión de información mediante la conmutación de paquetes. El crecimiento de esta red permitía a los usuarios ejecutar programas en modo remoto, además de proporcionar el servicio de correo electrónico a los investigadores de diferentes lugares geográficos. Más tarde ***ARPA*** propuso un proyecto llamado ***Internetting***, cuya finalidad era establecer conexiones entre distintas redes; sin embargo para su funcionamiento era necesario la presencia de un moderador que realizara la comunicación. Fue en 1974 cuando Robert Kahn y Vinton G. Cerf presentan TCP/IP (Transfer Communication Protocol / Interchange Protocol) entre redes, un protocolo de comunicación cuya función era la de controlar los

paquetes de información a transmitir. En esa época UNIX (desarrollado por los Laboratorios Bell en 1971) empezó a ser un sistema operativo muy importante. Entonces ARPA decidió contratar a la Universidad de Berkley para integrar TCP/IP al UNIX y así tener un sistema operativo orientado a redes de comunicaciones terrestres.

Más tarde empezaron a existir redes públicas en diferentes partes del mundo, por lo que la CCITT (actualmente *ITU*-International Telecomunication Union), decidió desarrollar el estándar ***X.25*** para promover una interfaz entre las redes públicas de conmutación de paquetes y sus clientes. El documento incluía una serie de propuestas sugeridas por Datapac (Canadá), Telnet y Tymet (Estado Unidos).

En ese tiempo surgió SNA (System Network Arquitecture), una arquitectura de redes de sistemas centrales (mainframes) de IBM, que permitió el acceso a enormes cantidades de información almacenadas en sistemas centrales. Posteriormente se presentó otra arquitectura nombrada como DECnet, creada para dar soporte a las comunicaciones sobre una variedad de redes incluyendo LAN de Ethernet y redes de banda básica y amplia.

Intel hace su aparición en el año de 1971 presentando el primer microprocesador, el 4004 de 4 bits. Éste y la versión perfecionada en forma de 8008 de 8 bits representa la base de la tecnología de los microprocesadores.

Más tarde surgió un invento que tuvo un gran impacto en el mercado de las computadoras fue la aparición de la computadora personal. Su diseño se debió a Steve Wozniak y junto con Steve Jobs, empezaron a fabricarlas y venderlas con el nombre de Apple II. Más tarde aparecería la competencia, IBM y Microsoft sacarían la computadora personal con sistema operativo MS-DOS.

Mientras tanto en el ámbito de las redes, DATAPOINT CORPORATION desarrolla un sistema de red ARCNet, basado en el método Token-Passing Lógico para controlar el acceso a red trabajando a una velocidad de 2.5 Mbps.

En 1980 se publican las especificaciones de Ethernet en el estándar DIX (Digital-Intel-Xerox) a 10 Mbps, el cual fue adaptado por el grupo de estándares de redes de área local del IEEE (Institute of Electrical and Electronic Engineers).

A su vez es publicado el estándar de Fax Group 3 por la CCITT especificando promedios de transmisión de hasta 9600 bps e incluía compresión incorporada que hacía posible transmitir una página típica en menos de 30 segundos.

En 1981, aparece Novell con el sistema Sharenet que más tarde se convertiría en Netware. Su propósito fue dar a las compañías una opción para cambiar su mainframe por una red de PCs. En este mismo año se crea la red BITNET (Because it's New Network) trabajando con el protocolo NJE (Network Job Entry) desarrollado por IBM, el cual permitió realizar el servicio de correo electrónico y la transferencia de archivos.

La Fundación Estadounidense para el Desarrollo de la Ciencia NSF decidió construir un ***backbone*** (red que funciona como espina dorsal) para la conexión de diversas redes regionales, utilizando como protocolo TCP/IP con una velocidad de transmisión de 56 Kbps. Esta colección de redes fueron los cimientos de Internet aunque no se le nombraba de ese modo.

En el año de 1985, IBM introduce al mercado la tecnología Token Ring basada en el concepto de utilizar una señal que circula alrededor de la red para dar a una unidad acceso a la misma. En ese año el IEEE presentó la norma 802.3 (Carrier Sense Multiple Acces with Collision Detection CSMA/CD; Acceso múltiple con detección de portadora y detección de colisiones) la cual determina el estándar en que se basa la tecnología Ethernet DIX a 10 Mbps.

Posteriormente, se crea el estándar ***SONET*** (Synchronous Optical Network, Red óptica sincrónica) con el objetivo de hacer posible la interconexión de redes de diferentes portadoras y unificar los sistemas digitales estadounidense, europeo y japonés.

El Instituto Tecnológico y de Estudios Superiores de Monterrey ITESM se conecta a BITNET en 1986, y la Universidad Nacional Autónoma de México lo hace en octubre del siguiente año. Para 1989, se instala el primer nodo Internet en México a través del enlace de ITESM hacia la Universidad de Texas en San Antonio UTSA, mediante una línea privada analógica de 4 hilos a 9600 bps.

El deseo de incrementar la velocidad de transferencia de datos en las redes, se ve manifiesto NSFNET cuando contrata a MCI (Microwave Communicatione Inc) para incluir en sus enlaces fibra óptica y aumentar la velocidad hasta 448 Kbps utilizando procesadores RISC IBM RS6000. Más tarde, las compañías Merit, MCI e IBM crean la compañía Advanced Network&Services Inc. ANS, que se encargaría de la red NSFNET (pública) y de la red comercial ANS CO+RE realizando enlaces de fibra óptica de hasta 45 Mbps.

En 1988, el ITU-T[1] estableció un estándar (I.122), que describía la multiplexación de circuitos virtuales en el nivel 2, conocido como el nivel de "frame" (trama). Esta recomendación fue denominada Frame Relay, ofreciendo una velocidad de transmisión de 2 Mps.

En el año de 1990 se da a conocer la especificación ***RFC*** 1157 que describe el protocolo SNMP (Simple Network Management Protocol), para proporcionar una manera sistemática de supervisar y administrar una red de cómputo.

En ese mismo año, fue puesto en funcionamiento el servicio SMDS (Switch Multimegabit Data Service), el primer servicio conmutado de banda ancha que se ofreció al público, operando a una velocidad de 45 Mbps.

En ese momento empezaban a surgir las nuevas tecnologías de redes de alta velocidad, no quedando atrás ATM (Asynchronous Transfer Mode, modo de transferencia asíncrono), siendo totalmente reconocido por su Forum creado en octubre de 1991 y cuyo propósito es el de incrementar el desarrollo de su tecnología.

En 1991 surge la Red Nacional Educativa y de Investigación, NREN (sucesora de NSFNET), cuya velocidad de operación se empezó a hablar de gigabits.

En 1992 se estableció la Sociedad Internacional ISOC para promover el conocimiento de Internet y de su tecnología. Para ese año comienza a operar MBone (Multicast Backbone) y su principal función es la de difundir audio y video en vivo en forma digital por todo el mundo a través de Internet.

En 1995 se anuncia oficialmente en México el Centro de Información de Redes (NIC-México) el cual se encarga de la coordinación y administración de los recursos de Internet asignados a México, tales como la administración y delegación de los nombres de dominio ubicados bajo .mx.

La tecnología Fast Ethernet a 100 Mbps es utilizada desde 1994, principalmente para aplicaciones multimedia y acceso a Intranet. Pero en el año de 1997, esta teconología se ve superada en velocidad al surgir Gigabit Ethernet a 1000 Mbps, utilizado para la transmisión de archivos grandes de imágenes y manejo de acceso a bases de datos.

Asimismo, SONET se definió como un sistema de fibra óptica de alta velocidad que provee una interfaz y un mecanismo para que sea posible la transmisión óptica de información digital, la cual soporta rangos de transmisión de 51.84 Mbps a 2.488 Gbps en la jerarquía de señal digital. Actualmente, el equipo de SONET es utilizado por una gran variedad de proveedores de servicio para transporte FDDI, ATM , incluyendo servicios de sistemas punto a punto y basados en anillo.

[1] Ver en el glosario ITU.

En 1999 surge la nueva dorsal de Internet (***Internet 2***), que fue gestada por varias Universidades de Estados Unidos y que les concede tener un canal de comunicación entre ellas de hasta 655 Mbps (OC-12). Como consecuencia surge el protocolo ***IPv6*** (Internet Protocol version 6; Protocolo Internet versión 6) que fue desarrollado por la IETF (Internet Engineering Task Force; Fuerza de trabajo de Ingeniería Internet) que tiene como característica un espacio de direcciones ampliadas y facilidades de encaminamiento mejoradas.

Actualmente, este tipo de tecnología que contribuye al crecimiento de las telecomunicaciones, trata de proporcionar mejores aplicaciones y servicios como por ejemplo, el comercio electrónico, la videoconferencia, los juegos, la telefonía, entre otros, a los usuarios que cada día son más

1.2. Análisis de estudio de las redes de computadoras.

Del anterior resumen podemos observar que el factor importante en la evolución de las redes de computadoras es el aumento de la velocidad de transmisión de datos ofreciendo grandes ventajas en el manejo de información. Por ejemplo, las organizaciones modernas suelen estar dispersas, y a veces tienen empresas distribuidas en varios puntos de un país o en el mundo, las cuales realizan intercambio de datos diariamente; por lo tanto, mediante una red es posible lograr que los programas y datos necesarios estén al alcance de los miembros de la organización. Otro aspecto importante que permite la interconexión de computadoras es la posibilidad de compartir recursos tales como impresoras u otros procesadores, para realizar tareas de procesamiento en menor tiempo.

Actualmente se están desarrollando muchas aplicaciones en las cuales se integran voz, video y gráficos, siendo sus archivos grandes volúmenes de información y por lo tanto no ofrecen portatilidad. Sin embargo, el desarrollo de la tecnología de las redes de banda ancha permite el envío de esos archivos, así como su ejecución en tiempo real. Esta tendencia se conoce como Desktop Multimedia Networking; es decir, la transmisión de aplicaciones multimedia a través de la red. En la tabla 1.1 se muestan algunas tecnologías que han surgido a través del tiempo. La característica más notable que diferencia unas de otras es la velocidad de transmisión con la que operan.

Año	Tecnología	Tipo de red	Velocidad	Medio físico	Aplicaciones
1969	ARPANet	WAN	8063 bits/seg	Línea telefónica	Conmutación de paquetes de almacén y envío
1973	ARPANet	WAN	56Kbps	Línea telefónica	Correo electrónico
1980	Ethernet	LAN o MAN	10Mbps	Cable coaxial	Compartición de archivos e impresoras
1980	ARCNet	LAN o MAN	2.5Mbps	Cable coaxial	Transferencia de datos en una red única.
1988	Frame Relay	WAN	2 Mps	Línea telefónica rentada	Aplicaciones con tráfico de voz y datos sobre una única red.
1990	SMDS	WAN	45Mbps	Línea telefónica rentada	Servicios de alta velocidad.
1991	ATM	WAN	(155- 622) Mbps	Canal de fibra óptica	Servicios de multimedia Servicios de telecomunicaciones: televisión de alta definición
1994	Fast Ethernet	LAN o MAN	100 Mbps	Cable de par trenzando, UTP, FTP	Multimedia y acceso a Internet
1997	Giga Ethernet	MAN	1 Gbps	Canal de fibra óptica 780 nm	Servicios de multimedia Acceso Bases de Datos
1997	SONET	WAN	51.84 Mbps 2.48 Gbps	Canal de fibra óptica	Video de alta resolución Redes LAN virtuales Emulaciones a través de un carrier Grandes canales de voz Servicios interactivos entre redes Ethernet Servicios de multimedia Desarrollo de aplicaciones remotas Multiconferencias en tiempo real

Tabla 1.1 Tecnologías de redes de computadoras.

Con relación a la tabla 1.1, en la gráfica 1.1 podemos apreciar con mayor claridad el incremento en la velocidad de transmisión de datos en las redes WAN.

A partir de 1980 la velocidad se elevó hasta 2 Mbps y para 1990 se incrementó más de cuatro veces. Pero el incremento más notable fue a partir de 1991, desde una velocidad de 622 Mps se llegó hasta 2.8 Gbps. Podemos decir que lo ocurrido tiene mucho que ver con el empleo de nuevas tecnologías de hardware (cableado, procesadores,

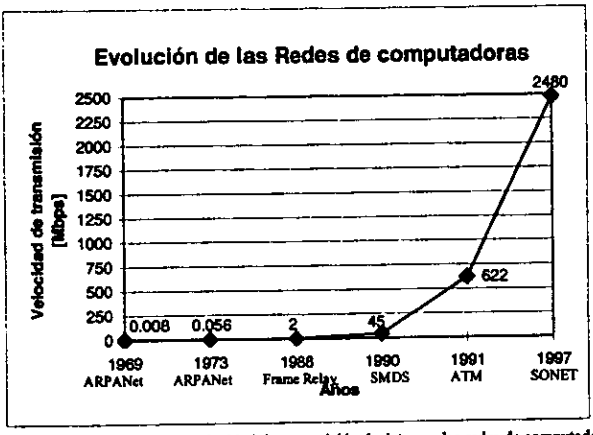


Figura 1.1. Incremento de la velocidad de transmisión de datos en las redes de computadoras.

encaminadores, etc.) y, del mejoramiento y creación de software (protocolos de comunicación, sistemas operativos, etc.) empleados. Además, se observa que la velocidad de transmisión no ha aumentado desde el año de 1997, pues parece que hasta este momento este tipo de tecnología satisface una de las expectativas importante (transferencia de datos rápida). Sin embargo, es posible que aparezcan nuevas ofertas aunque primero se debería trabajar en otro campo importante que es el software para poder tener una transferencia eficiente donde se incluyera la integridad y seguridad de los datos.

1.3. Seguridad de la información.

Actualmente las redes de comunicaciones son la vía principal para el envío y recibo de datos los que suelen ser atractivos a personas ajenas. Por ejemplo, en las instituciones bancarias los clientes requieren de un número identificador para realizar sus transacciones, si éste cayera en manos extrañas podría llevarse a cabo un gran robo. De igual forma, en compañías comerciales existe información tales como los costos y precios, investigaciones de mercado, planes estratégicos, listas de clientes etc., que puede ser de gran interés para un competidor. También la información científica o técnica que maneja una institución educativa o militar puede ser usada para realizar daños a la sociedad. Es por eso que las compañías comerciales, instituciones educativas, bancarias o de carácter personal han estado utilizando ciertos medios para proteger la información que manejan.

la sociedad. Es por eso que las compañías comerciales, instituciones educativas, bancarias o de carácter personal han estado utilizando ciertos medios para proteger la información que manejan.

La seguridad de la información puede remontarse al año 2000 a. C, cuando los egipcios emplearon en su escritura jeroglíficos para codificar la información. Pero no fueron los únicos en emplear esta técnica, también las civilizaciones de Babilonia, Mesopotamia y Grecia tuvieron la necesidad de ocultar mensajes.

La codificación de la información fue realizada por primera vez por Julio César, y posteriormente en las dos guerras mundiales. Una de las mejores máquinas de codificación fue la máquina alemana Enigma, cuyo propósito era generar mensajes codificados para los alemanes durante la Segunda Guerra Mundial. Más tarde, Alan Turing pudo concebir las primeras computadoras para descifrar mensajes: los Colosus y el proyecto Ultra.

Durante los años cuarenta y cincuenta, los centros de información fueron estructurados como lugares impenetrables preocupándose únicamente por la seguridad física. Pero al relacionarse la criptología y la computación durante la Segunda Guerra Mundial hubo un cambio de enfoques; la seguridad de la información se relacionaba directamente con los datos y no sólo con los dispositivos que los almacenaban o los muros del cuarto donde se encontraban.

En esa época, Shannon mostró un tipo de cifrado "perfecto" llamado cifrado de Vernam, sin embargo la desventaja que presentó fue la de ser muy caro y difícil de implementar porque utiliza claves tan grandes como el mensaje secreto. Estas claves se conocían como libretas de un sólo uso o combinación con series aleatorias.

En 1960, John Conway desarrolló "living" un software que pudo replicarse asimismo. La idea de estos programas se expandió en la comunidad académica e investigadora y los estudiantes empezaron a crearlos como parte de un desafío pero en forma inofensiva. Más tarde este tema ganó terreno en las investigaciones del ***MIT*** y Xerox's PARC pero fueron mantenidas en secreto.

En los primeros años de la década de los setenta, la criptología empieza a ser considerada como una industria y una tecnología estratégica con muy importantes aplicaciones civiles. Los criptosistemas se empezaban a diseñar mediante el empleo de una clave de cifrado, la cual, también era utilizada para descifrar el mensaje. En 1976 dos investigadores de la Universidad de Stanford, Diffie y Hellman propusieron que ambas claves fueran diferentes y que la clave de descifrado no pudiera derivarse de la clave de cifrado. Este tipo de criptografía se le llamó de clave pública porque se requería que cada usuario tuviera dos claves: una clave pública usada por todo el mundo para cifrar los mensajes, y una clave privada para descifrarlos.

Para el año de 1977, la National Security Agency e IBM desarrollan el proyecto "Lucifer", creando el cifrado Data Encryption Standard (DES). Este estándar empezó a ser utilizado en la industria.

Más tarde, se crea un algoritmo de clave pública llamado RSA, que significa las iniciales de sus tres descubridores Rivest, Shamir y Adleman miembros del MIT. Este método se basa en ciertos principios de la teoría de números, garantizando que la información cifrada con una clave pública sólo puede ser descifrada con la clave privada.

En 1980 se desarrolló en el MIT, Kerberos para permitir a los usuarios de estaciones de trabajo el acceso a recursos de una manera segura.

Con el objetivo de administrar los nombres de las máquinas se crea DNS (Domain Naming System) para relacionar las direcciones de los host y destinos de correo electrónico con las direcciones IP.

Al mismo tiempo, el crimen computacional se incrementa. Los ***hackers*** irrumpen en cuentas privadas y transferencias ilegales. Pero en 1980 con el advenimiento de las PCs, los virus finalmente entraron como una real

amenaza. Uno de los primeros virus documentados fue escrito en 1983 por Fred Cohen, un estudiante de la Universidad de California del Sur. El programa que escribió tenía la característica de que al ser instalado en el disco duro podía replicarse él mismo hasta destruir el sistema de la computadora. En 1985 se ofreció al público uno de estos programas a través de boletines electrónicos. Pronto los programas incluirían el ***Caballo de Troya*** NUKE-LA y EGABTR, así como un número de virus tradicionales.

En 1985 el Departamento de la Defensa de Estados Unidos crea el "Libro Naranja" con el propósito de establecer un estándar de niveles de seguridad para proteger de ataques al hardware, al software y a la información almacenada.

A pesar de esto, en el año de 1990 los llamados hackers hacen una aparición inesperada dentro de organismos importantes, entre ellos cabe mencionar: El Pentágono, la NASA, los laboratorios de Berkeley, y AT&T.

En 1991 el NIST (National Institute of Standards and Technology) propone el uso de una variación del algoritmo de clave pública de El Gammal (dicho algoritmo se basa en el cálculo de logaritmos discretos), para crear DSS (Digital Signature Standard), un nuevo estándar de firmas digitales. Sin embargo, surgieron diversos ataques criptoanalíticos y junto con la falta de credibilidad técnica y política a este sistema hicieron naufragar el proyecto. El DSS se criticó por ser, secreto, lento (10 a 40 veces más lento que el RSA para comprobar firmas) e inseguro (clave fija de 512 bits).

Con el tiempo, el uso del correo electrónico E-Mail (Electronic Mail, Correo electrónico) se convirtió en la forma de comunicación más utilizada, sin embargo, este servicio ofrecía la desventaja de ser fácil de interceptar. Ante esta situación en 1991 Zimmerman desarrolla un programa conocido como PGP (Pretty Good Privacy o Intimidad Bastante Buena), el cual consiste en la codificación adaptable a la correspondencia electrónica basado en los algoritmos RSA, MD5 e IDEA. Después de tres años de su lanzamiento público fue copiado por miles de usuarios y fue entonces cuando comenzaron los problemas para Zimmermann. El programa PGP había sido encontrado en las computadoras de los traficantes de drogas y de los mafiosos que lo utilizaban para codificar sus cuentas. A consecuencia de esto Zimmerman podía ser condenado a cuatro años de cárcel.

El gobierno de Clinton intentó contrarrestar esta tendencia imponiendo el desarrollo del Clipper Chip, un circuito que contiene un algoritmo de cifrado llamado Skipjack. Sin duda lo que sorprendió con este nuevo desarrollo fue la declaración tan abierta que se hizo: "...las autoridades policiacas y judiciales podrán descifrar, sin permiso de emisor o receptor, las comunicaciones telefónicas y telemáticas mediante un depósito de claves (key scrow system)". De esta forma, los agentes del FBI y los funcionarios del Departamento de Justicia podrían investigar delitos a pesar de que los presuntos criminales codifiquen con el Clipper Chip sus conversaciones, faxes y mensajes de correo electrónico. Pero en el momento que iban a reproducirse en serie para incluirlos en los futuros teléfonos, se descubrió que tenía un defecto susceptible de hacerlo inviolable incluso para el Gobierno, por lo que se pospuso su liberación.

Agencias de seguridad federal han estado trabajando junto con las compañías de telecomunicaciones en el diseño de puertas traseras (backdoors) dentro de la Infraestructura de Información Nacional (NII- National Information Infrastructure), también conocida como la "superautopista de información nacional". Con esto, el gobierno tendrá la habilidad de grabar el tiempo, el origen, el receptor de cada llamada y el mensaje electrónico (e-mail) escuchando las comunicaciones privadas continuamente sin tener autorización del usuario. En otras palabras, cualquier sistema o red de computadoras que esté conectada al NII, se le solicitará que incluya una puerta trasera para que de esa manera las agencias gubernamentales como el FBI, puedan examinar la información. Cualquier información que pase a

través de la red telefónica nacional hasta la red local de computación llegaría a ser examinada por el gobierno, incluyendo conversaciones telefónicas, video, imágenes y texto. El gobierno federal asegura que estas "puertas traseras" sólo serán usadas para capturar a criminales y la privacidad será protegida. Sin embargo, se descubrió antes de tiempo una red de espionaje mundial surgida tras la guerra fría, resultado de complejas negociaciones entre los países miembro de la OTAN y que fueron absolutamente secretas hasta que una filtración en 1997 de un documento de la Unión Europea permitió a la opinión pública conocer la existencia de ésta, conocida como el ***ECHELON.***

A principios de 1998 hubo un ataque al Pentágono pasando por alto sistemas ***firewalls***. Estos sistemas de seguridad se consideraron demasiado pobres ya que pudieron ser accesado por dos quinceañeros. John Hamre, número dos del Departamento aseguró que habían entrado "en mayor o menor medida" en las redes de once centros militares (siete de la Fuerza Aérea y cuatro de la Marina). Posiblemente también penetraron en docenas de computadoras con información gubernamental, incluyendo laboratorios de investigación de armamento nuclear. Al parecer, los adolescentes utilizaron Netdex un proveedor de acceso a Internet de Santa Rosa (California), mediante el cual intentaban acceder a las redes militares sirviéndose también de centros universitarios de investigación.

Se puede apreciar que la tendencia de la mayoría de las empresas es la de proteger su información, que viaja a través de las redes de computadoras, por medio de la criptografía. La utilización de ella ha resultado más convincente que otras y es por eso que se está implementando tanto en hardware como en software. Sin embargo, diseñar algoritmos de cifrado es muy difícil, por lo que es más conveniente estudiar aquellos que ofrecen mayor robustez para implementarlos.

1.4. Amenazas deliberadas sobre datos informáticos.

Se entiende por amenaza una condición del entorno del sistema de información (que incluye personas, máquinas, sucesos o ideas) que, dada una oportunidad, podría dar lugar a que se produjese una violación de la seguridad (en lo referente a confidencialidad, integridad, disponibilidad o uso legítimo). Se ha considerado cuatro categorías de amenazas o ataques y son las siguientes:

1. *Interrupción.* Un recurso del sistema es destruido o se vuelve no disponible. Este es un ataque contra la disponibilidad. Ejemplos de este ataque son la destrucción de un elemento hardware, como un disco duro, cortar una línea de comunicación o deshabilitar el sistema de gestión de archivos.
2. *Intercepción.* Una entidad no autorizada se apodera de un conjunto de datos antes de que llegue a su destino. Este es un ataque contra la confidencialidad. La entidad no autorizada podría ser una persona, un programa o una máquina.
3. *Modificación.* Una entidad no autorizada no sólo consigue acceder a un recurso, sino que es capaz de manipularlo e insertar objetos en él. Este es un ataque contra la integridad. Ejemplos de este ataque son la modificación de valores en un archivo de datos, alterar un programa para que funcione de forma diferente y modificar el contenido de mensajes que están siendo transferidos por la red.

Estos ataques se pueden asimismo clasificar de forma útil en términos de ataques pasivos y ataques activos.

Ataques pasivos

En los ataques pasivos el atacante no altera la comunicación, sólo la escucha o monitorea; para obtener la información que está siendo transmitida. Sus objetivos son la intercepción de datos y el análisis de tráfico. Una técnica más sutil para obtener información de la transmisión, consiste en:

- Obtención del origen y destinatario de la comunicación, leyendo las cabeceras de los paquetes monitoreados.
- Control del volumen de tráfico intercambiado entre las entidades monitoreados, obteniendo así información acerca de actividad o inactividad.
- Control de las horas habituales de intercambio de datos entre las entidades de la comunicación, para extraer información acerca de los períodos de actividad.

Un ejemplo de ataque pasivo es el conocido como trap doors, que son puntos de entrada fáciles al software, usualmente puestos por el diseñador del sistema quien los usa para examinar o monitorear el programa sin necesidad de revisar toda la seguridad que ha sido implementada. El diseñador del sistema permite el acceso extenso al sistema usando una palabra clave, en lugar de pasar por diferentes capas de seguridad que son implementadas en el programa principal. El peligro en los traps doors es que al ser creados para el uso del diseñador del sistema, también ofrece el poder de acceso a un intruso.
Los ataques pasivos son muy difíciles de detectar, ya que no provocan ninguna alteración de los datos. Sin embargo, es posible evitarlos mediante el cifrado de la información y otros mecanismos que se verán más adelante.

Ataques activos

Estos ataques implican algún tipo de modificación del flujo de datos transmitido o la creación de un falso flujo de datos, pudiendo subdividirse en cuatro categorías:

- *Suplantación de identidad.* El intruso se hace pasar por una entidad diferente. Normalmente incluye alguna de las otras formas de ataque activo. Por ejemplo, secuencias de autenticación pueden ser capturadas y repetidas, permitiendo a una entidad no autorizada acceder a una serie de recursos privilegiados suplantando a la entidad que posee esos privilegios, como al robar la contraseña de acceso a una cuenta.
- *Reactuación.* Uno o varios mensajes legítimos son capturados y repetidos para producir un efecto no deseado, como por ejemplo ingresar dinero repetidas veces en una cuenta dada.
- *Modificación de mensajes.* Una porción del mensaje legítimo es alterada, o los mensajes son retardados o reordenados, para producir un efecto no autorizado.
- *Degradación fraudulenta del servicio.* Impide o inhibe el uso normal o la gestión de recursos informáticos y de comunicaciones. Por ejemplo, el intruso podría suprimir todos los mensajes dirigidos a una determinada entidad o se podría interrumpir el servicio de una red inundándola con mensajes espurios. Entre estos ataques se

encuentran los de negación de servicio, consistentes en paralizar temporalmente el servicio de un servidor de correo, Web, FTP, entre otros.

Otro tipo de ataques activos se puede realizar mediante la transmisión de virus. Los virus son programas que tienen la capacidad de reproducirse y propagarse, es decir, son capaces de hacer una o más copias de sí mismos transfiriéndose a otros disquetes y a otras computadoras, además de hacer esta copia de una forma que resulta transparente al usuario.

Los virus durante el proceso de réplica o propagación, pueden sufrir mutaciones (modificaciones en el código programado originalmente), tienden a destruir (borrar o alterar) la información contenida en un disquete o en el disco duro de una computadora.

Existen diferentes categorías en las cuales se agrupan los virus de acuerdo a sus características. Una de las más generales es la que se presenta a continuación:

- *Caballos de Troya.* Son aquellos que se introducen al sistema bajo una apariencia totalmente diferente a la de su objetivo final; esto es, que se presentan como información perdida o basura, sin ningún sentido. Pero al cabo de algún tiempo, y esperando la indicación programada, comienzan a ejecutarse y a mostrar sus verdaderas intenciones. En general, estos virus son destructores de la información contenida en los discos, ejemplo de ellos son 12-Tricks y AIDS.
- *Bombas de tiempo.* Son programas ocultos en la memoria del sistema, en los discos o en los archivos de programas ejecutables (con extensión .COM o .EXE). Esperan una fecha u hora determinadas para "explotar". Algunos de éstos no son destructivos y sólo exhiben mensajes en la pantalla al llegar el momento de la explosión. Llegado el momento, se activan cuando se ejecuta el programa que los contiene.
- *WORM.* Un gusano (worm) es un programa que cambia y destruye datos como los virus pero también viajan y dañan de computadora a computadora a través de la red; por ejemplo Internet. Un gusano, después de viajar del sistema de una computadora a otra, puede hacer muchas copias de si mismo, gastando enormes cantidades de tiempo de cómputo, y puede incluso parar una red de computadoras.

1.5. Niveles de Seguridad.

Los mecanismos de seguridad pueden ser implantados en dispositivos (hardware), programas (software) o sistemas de seguridad (dispositivos y aplicaciones en conjunto). La elección de mecanismos de seguridad depende de qué tan importante es la información que se maneja. Se han establecido criterios de niveles de seguridad para la evaluación de sistemas entre ellos se encuentran TCSEC (Trust Computer System Evaluation Criteria) en Estados Unidos, CTCPEC (Canadian Trusted Computer Product Evaluation Criteria) en Canadá e ITSEC (Information Tecnology Security Evaluation) el cual es un acuerdo entre países europeos: Alemania, Inglaterra, Francia y Holanda. Sin embargo, cada uno de ellos presenta diferencias entre sus niveles de seguridad.

Nivel	TCSEC	Nivel	CTCPEC	Nivel	ITESEC
D1	El sistema entero no es confiable. No existe protección para el hardware, ni autentificación respecto de los usuarios. No hay control en el manejo de la información de cada usuario.	EAL-1	Se basa en el análisis de las funciones de seguridad del producto. Sólo se requiere de los diseños funcionales y de interfaz del producto para comprender el comportamiento de seguridad. *Se le llama producto a las aplicaciones o dispositivos de seguridad; así como la combinación de ellos.	E0	El sistema no ofrece ninguna garantía de seguridad. No existe software ni hardware que garantice la integridad de los programas y datos.
C1	Los productos de seguridad son programas que proporcionan interfaces de identificación de usuarios que requieren de contraseñas (passwords) para tener acceso a algunos programas y el manejo de la información dentro del sistema.	EAL-2	Se realiza un análisis de las especificaciones funcionales y de interfaz, junto con una revisión de diseño de los subsistemas del producto	E1	Los productos de seguridad deben tener una descripción informal del diseño de su arquitectura. Para verificar la seguridad especificada basta con aplicar un examen funcional.
C2	Además de las funciones del nivel C1, se incluyen niveles de autorización. Además, el sistema debe ser auditado registrando las acciones del administrador y usuarios.	EAL-3	Se requiere de una fuente externa para validar la seguridad. En este nivel el producto se ha diseñado teniendo en mente la seguridad y no después de la etapa de diseño.	E2	Debe existir una descripción informal del diseño detallado. Las pruebas de su examen funcional serán evaluadas. Debe tener un sistema de control de configuración y un proceso de distribución aprobatorio
B1	Se refiere a la protección de seguridad etiquetada con soporte de seguridad multinivel.	EAL-4	El producto está diseñado, aprobado y revisado metódicamente, basado en prácticas razonables de desarrollo de software comercial.	E3	Se realiza la evaluación de los diseños del código fuente o del hardware del producto.
B2	La protección estructurada requiere que todos los objetos estén etiquetados. Los dispositivos como discos, cintas y terminales, pueden tener asignado uno o varios niveles de seguridad	EAL-5	El desarrollador debe aplicar prácticas de desarrollo de software comercial, así como técnicas especializadas de ingeniería de seguridad. Se debe presentar las especificaciones de diseño y la forma en que dichas especificaciones se implementan de modo funcional en el producto.	E4	Existe un modelo formal de soporte de la administración de seguridad del producto. Las funciones de la seguridad garantizada, el diseño de la arquitectura y el diseño detallado, serán especificados en un estilo semiformal.
B3	Dominios de seguridad refuerza los dominios con la instalación de hardware	EAL-6	Consta de un diseño de verificado semiformal y de un componente de prueba: incluye los anteriores niveles, con el requerimiento adicional de una presentación estructurada de la implantación. En este nivel asegura que todo el ciclo de diseño exista un proceso de desarrollo estructurado, controles de desarrollo y controles de manejo de configuración.	E5	Debe existir una correspondencia entre el diseño detallado y el código fuente y/o los diseños de hardware
A1	Proceso estricto de diseño, control y verificación. Incluyendo los niveles inferiores, el diseño debe verificarse matemáticamente, y debe realizarse un análisis de los canales cubiertos y de distribución confiable	EAL-7	Este nivel consiste en una revisión independiente y formal del diseño, con diseño verificado y una etapa de pruebas. Se debe probar cada faceta del producto, buscando puntos vulnerables obvios o no, que después son verificados por una fuente independiente.	E6	Las funciones de la seguridad garantizada y el diseño de la arquitectura serán especificadas en un estilo formal, consistente con la especificación del modelo formal de la administración de seguridad.

Tabla 1.2. Criterios de niveles de seguridad.

En la tabla 1.2, se puede observar que los primeros niveles de seguridad de los criterios ITSEC y CTCPEC se enfocan en la evaluación del producto, verificando que la seguridad sea garantizada desde la etapa de diseño.

Los criterios del ITSEC fueron desarrollados a partir de los esquemas de seguridad de la tecnología de información europea. Los sistemas y productos tienen más libertad en el diseño de la arquitectura pero tienen más limitaciones en cuanto a sus aplicaciones.

Los criterios del TCSEC consideran importante el diseño del producto hasta los últimos niveles, pero otro aspecto a considerar es la evaluación de servicios y aplicaciones, por lo que se debe estudiar las técnicas de seguridad más convenientes para lo que se desea proteger.

En México existe una situación preocupante con respecto a la seguridad, ya que parece que proteger la información manejada en las redes de comunicaciones no es de mucha importancia, incluso no existe ninguna ley que penalice los delitos electrónicos. Ante esta situación algunas instituciones u organismos principalmente los bancos, han recurrido a mecanismos como el cifrado para garantizar que las transacciones realizadas no sean alteradas. Es necesario que tanto empresas gubernamentales como privadas establezcan sistemas de seguridad, pero ¿cómo evaluar la garantía que ofrecen sino existe un criterio de seguridad en las políticas de informática en México?

Nivel	Descripción
N0	No existe ninguna forma de seguridad por lo que no es confiable el sistema entero. No hay dispositivos ni programas que limiten a usuarios ajenos el uso del sistema. Un ejemplo de este nivel es la instalación de un sistema que no es capaz de monitorear o detectar el usuario que esta operando.
N1	Se emplean sistemas de autentificación como por ejemplo palabras clave (password) que sólo conoce el usuario y determina los derechos de acceso a programas e información que tiene dentro del sistema, así como también la ejecución de ciertos comandos. Cada usuario puede proteger su información mediante el establecimiento de los permisos que especifique. Sin embargo el control de este sistema lo tiene el administrador el cual puede acceder a todos los programas e información sin existir una vigilancia en sus acciones. La evaluación de las aplicaciones que se encuentra en esta categoría será aprobada cuando las tareas que especifican sean cumplidas durante su ejecución en un sistema comercial.
N2	Se incluye el nivel uno y la creación de niveles de autorización en el sistema. El sistema debe manejar una seguridad multinivel, no todos los usuarios tienen derecho a ejecutar ciertos comandos, o que el administrador pueda hacer uso con toda libertad de la información perteneciente a usuarios del sistema. En este último caso interviene el sistema de auditoría que registra tanto las acciones de los usuarios como el administrador que está a cargo del control de la red. En este nivel todavía se habla de aplicaciones, considerando más restricciones de seguridad. Para hacer efectivas las técnicas de seguridad, éstas deben implantarse desde el momento del diseño de la aplicación y evaluarlas para garantizar su nivel.
N3	Además de dejar a cargo el monitoreo y control de las acciones de usuarios al sistema, existe otro mecanismo de protección para el manejo de información como lo es la codificación.
N4	Contiene los anteriores niveles y la implementación de dispositivos, por ejemplo: los firewalls, conocidos también como cortafuegos. Estos dispositivos solamentente filtran la información, deciden que servicios pueden ser accesados desde el exterior de una red privada, por quienes pueden ser ejecutados y también que servicios pueden hacer uso los usuarios del exterior.
N5	En este nivel se debe pensar en seguridad de dominios mediante la instalación de sistemas de seguridad. El sistema debe especificar detalladamente su funcionalidad, diseños de hardware y debe contener funciones de prueba que lo validen.
N6	Es el nivel de seguridad más efectivo pero el más costoso, puesto que se necesita implantar sistemas de seguridad cuyo análisis debe considerar todo el sistema que se requiere asegurar. Para su evaluación se requiere del diseño de la arquitectura, el diseño de implantación, documentos de su funcionalidad, las garantías que establece. Debe probarse su funcionalidad y a su vez buscar los puntos vulnerables que pudiera tener.

Tabla 1.3 Propuesta de los niveles de seguridad de acuerdo a los criterios establecido por Estados Unidos, Canadá y los países europeos.

En la tabla 1.3 se hace una propuesta para establecer un criterio de seguridad en nuestro país. Este consiste en siete niveles, los cuales son una recopilación de las características de los criterios de seguridad de Estados Unidos, Canadá y el Continente Europeo, con el fin de tener un criterio que englobe aspectos de seguridad del software y del hardware.

En la actualidad existen en el mercado diferentes productos de software y hardware que proporcionan seguridad en los datos que se manejan en la red de una empresa o de un usuario individual, en el capítulo 4 se describen algunos de ellos.

Capítulo 2

Redes de Computadoras

Las redes de computadoras han tenido un gran impacto en la sociedad ya que han permitido agilizar el trabajo de las empresas y de los usuarios individuales de tal forma que han impactado grandemente factores de eficiencia y productividad. Sin embargo, su implementación involucra un alto grado de conocimiento de las tecnologías empleadas, topologías, protocolos de comunicación y medios físicos, entre otros. Por ello a continuación se mencionan algunos conceptos que describen los componentes principales que intervienen en el diseño, instalación y buen funcionamiento de una red.

2.1. Estructura de la red de comunicaciones.

Una red de comunicaciones puede ser un sencillo sistema constituido por dos equipos terminales de datos (***ETDs***), tales como computadoras personales, estaciones de trabajo, cajeros automáticos, etc. Para que exista un intercambio de datos entre estos equipos se requiere de una línea de transmisión también conocida como ***canal físico***. Y de un ***canal lógico***, de tal manera que los datos enviados de un equipo terminal lleguen a su destino mediante la dirección del equipo a donde se desean enviar. Otro dispositivo importante es el equipo de comunicaciones de datos (***ETCD***), el cual tiene la finalidad de conectar los equipos terminales a la línea o canal de comunicaciones (por ejemplo el modem).

Para que un ETD pueda accesar a una aplicación ejecutada por otro ETD se requiere de interfaces que permitan la interacción entre equipos proporcionadas por el protocolo empleado, y un canal físico. Los protocolos son reglas y convenciones establecidas para determinar la forma de comunicación entre los equipos terminales y los de comunicación, que establecen: quién debe hacer, qué se debe hacer y cuándo se debe de hacer algo. En la figura 2.1 podemos ver un sencillo sistema de comunicación de datos.

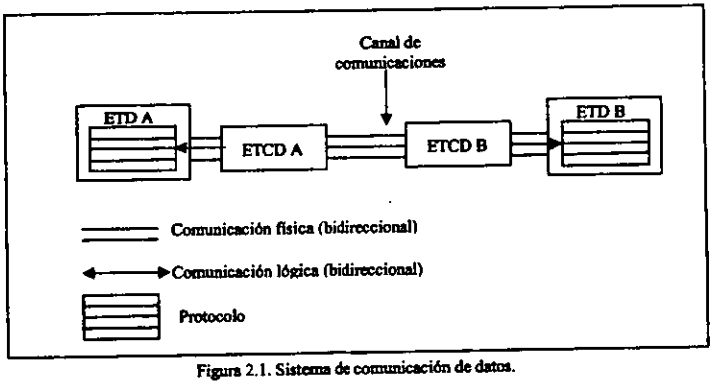


Figura 2.1. Sistema de comunicación de datos.

2.2. Transmisión.

El intercambio de datos entre los equipos terminales y los de comunicación suelen llevarse a cabo mediante uno de los tres siguientes tipos de transmisión:

- **Símplex.** Transmisión en un solo sentido (Figura 2.2.a).
- **Semidúplex.** Transmisión en ambos sentidos, pero sólo en uno en cada momento (también llamada bidireccional alternada (Figura 2.2.b)).
- **Dúplex integral** (o dúplex). Transmisión en ambos sentidos a la vez, también es llamada bidireccional simultánea (Figura 2.2.c).

La transmisión en modo símplex es habitual en televisión y en radiodifusión comercial. En comunicación de datos no es tan frecuente pero existen algunas aplicaciones en las que se emplea la comunicación símplex, como la telemetría. Existen aplicaciones del tipo pregunta/respuesta, en las cuales un equipo terminal envía una pregunta a otro equipo terminal y queda a la espera de que el proceso de aplicación obtenga la respuesta o la calcule (o ambas cosas), y devuelva el resultado. Los sistemas basados en terminales (terminales con teclado y terminales con pantalla de vídeo), suelen usar técnicas semidúplex. El dúplex integral permite transmitir en ambas direcciones a la vez, sin estar sometido a la estructura de parada y espera del semidúplex. Los sistemas dúplex son muy utilizados en las aplicaciones que exigen un empleo constante del canal, un elevado caudal de tráfico y un tiempo de respuesta rápido.

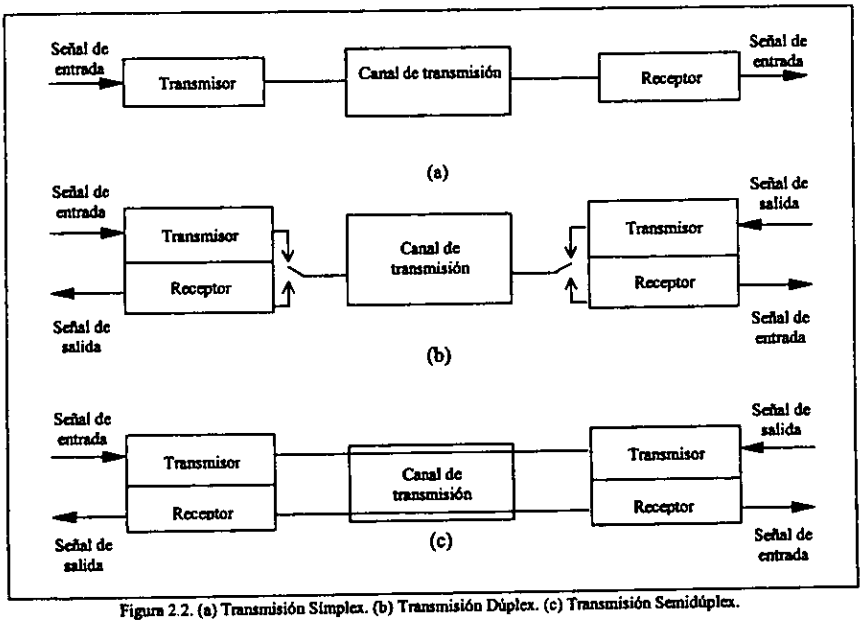


Figura 2.2. (a) Transmisión Símplex. (b) Transmisión Dúplex. (c) Transmisión Semidúplex.

En telefonía se utilizan los términos pares y cuadretes para describir el circuito que compone el canal. Los circuitos de pares suelen ser circuitos semidúplex constituidos por dos hilos, uno de ellos sirve para transmitir los datos, y el otro es la línea de retorno eléctrico[2]. Los circuitos de cuatro hilos, o circuitos de cuadretes mostrados en la figura 2.3, suelen ser circuitos dúplex constituidos por dos pares de dos hilos cada uno; dos de los hilos transmiten los datos y los otros dos cierran los correspondientes circuitos. Para las compañías telefónicas, un enlace de dos hilos suele

[2] El retorno eléctrico tiene como finalidad el cierre de flujo de corriente. Anteriormente, se empleaba la tierra como retorno eléctrico; sin embargo la calidad de la señal era muy pobre, al incluir la línea la calidad de la señal mejoró pero existía interferencia en ambos hilos por lo que se trenzaron los pares para eliminar ese efecto.

corresponder a un circuito telefónico conmutado normal, mientras que un circuito de cuatro hilos suele ser una ***línea alquilada***, no conmutada.

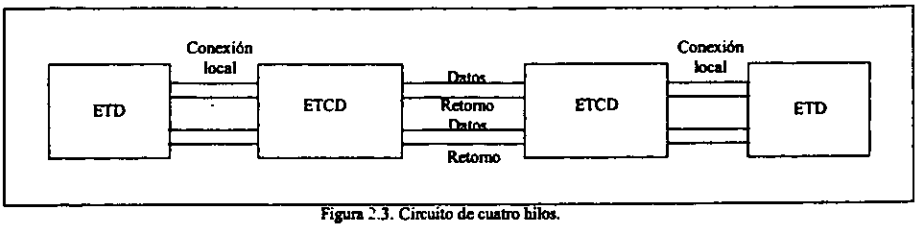


Figura 2.3. Circuito de cuatro hilos.

La función del conmutador (*ECD*) es encaminar el tráfico hasta su destino final a través de la red evitando los dispositivos y canales ocupados o fuera de servicio. Asimismo, el conmutador puede dirigir los datos hacia su destino final a través de componentes intermedios, que pueden ser a su vez, otros equipos de conmutación. Observe la figura 2.4.

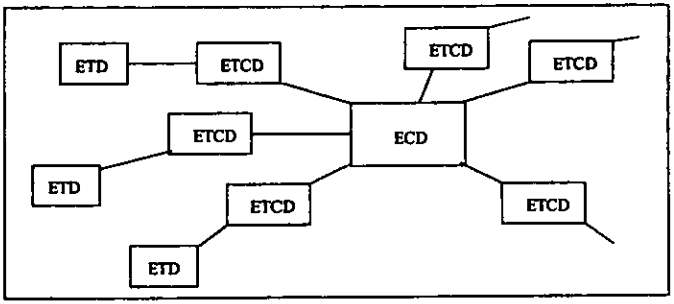


Figura 2.4. Equipo de conmutación de datos.

2.3. Topologías de red.

La configuración de una red suele conocerse como topología, es decir, la forma en que se conecta físicamente la red. Para establecer la topología de una red, se deben considerar los siguientes aspectos:

- Fiabilidad.
- Costo.
- Respuesta óptima y un caudal eficaz máximo.

La fiabilidad de una red se refiere a la capacidad que tiene la misma para transportar datos correctamente (sin errores), de un equipo terminal a otro, incluyendo también la capacidad de recuperación de errores o datos perdidos en la red, ya sea por fallo del canal, el equipo terminal o el de comunicaciones. La fiabilidad está relacionada

también con el mantenimiento del sistema, cuando un componente crea problemas, el sistema de diagnóstico de la red ha de ser capaz de identificar y localizar el error, aislar la falla y si es preciso, aislarlo del resto de la red.

Para encontrar el camino más económico entre el emisor y receptor dentro de la red se podría minimizar la longitud real del canal que une los componentes, lo cual suele implicar el encaminamiento del tráfico a través del menor número posible de componentes intermedios; o bien, proporcionar el canal más económico para cada actividad concreta, por ejemplo, transmitir los datos de baja prioridad a través de un enlace de baja velocidad por línea telefónica normal, lo cual es más barato que transmitir esos mismos datos a través de un canal vía satélite de alta velocidad.

El reducir al mínimo el tiempo de respuesta implicaría disminuir el retardo entre la transmisión y la recepción de los datos de un equipo terminal a otro. El caudal eficaz expresa la cantidad máxima de datos de usuario que es posible transmitir en un determinado período de tiempo.

Ahora bien, veamos a continuación la descripción de algunas topologías de red más comunes.

2.3.1. Topología jerárquica

En la topología jerárquica existe un punto de concentración de las tareas de control y de resolución de errores llamado nodo central. En la mayoría de los casos este nodo es el equipo terminal situado en el nivel más elevado de la jerarquía el cual controla otros equipos terminales de nivel inferior y éstos a su vez tienen control sobre otros equipos terminales subordinados, reduciendo así la carga de trabajo del equipo terminal de mayor jerarquía.

El nodo central suele ser una gran computadora central (mainframe) que controla todo el tráfico entre los distintos equipos terminales. Este hecho no sólo puede crear saturaciones de datos, sino que además plantea serios problemas de fiabilidad. En caso de que falle esa computadora central, toda la red dejará de funcionar a no ser que exista otra computadora sustituta que realice todas las funciones de la computadora central averiada.

Las redes con topología jerárquica se conocen también como redes verticales o en árbol. La palabra "árbol" es debido a la semejanza que tiene la estructura con un árbol, cuyas ramas van abriéndose desde el nivel superior hasta el más bajo. Observe la figura 2.5.

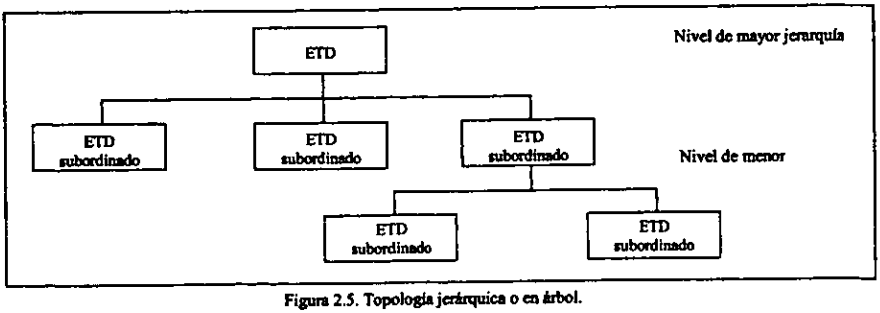


Figura 2.5. Topología jerárquica o en árbol.

2.3.2. Topología en bus.

Esta estructura es frecuente en las redes de área local. El bus permite que todas las estaciones y sus correspondientes interfaces conectadas a lo largo del canal reciban todas las transmisiones. La principal limitación que presenta la topología en bus es que existe un sólo canal de comunicaciones para todos los dispositivos de la red y si éste falla toda la red deja de funcionar. Algunos fabricantes proporcionan canales completamente redundantes por si falla el canal principal, y otros ofrecen conmutadores que permiten rodear un nodo en caso de que falle. Otro inconveniente de esta configuración estriba en la dificultad de aislar los fallos de un dispositivo particular conectado al bus. La falta de puntos de concentración complica la resolución de este tipo de problemas. Para una mejor referencia, observe la figura 2.6.

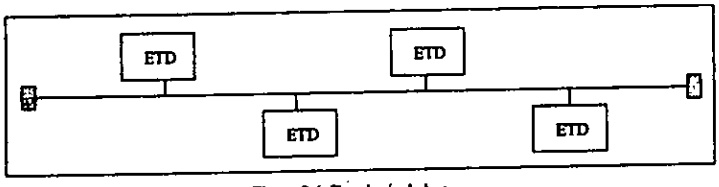


Figura 2.6. Topología de bus

2.3.4. Topología en estrella

La topología en estrella es una de las más empleadas en los sistemas de comunicación de datos. En este tipo de topología, todo el tráfico se origina en el núcleo de la estrella; es decir en el nodo central. Dicho nodo se encarga del control total de los equipos terminales conectados a él y es responsable de encaminar el tráfico hacia el resto de los componentes, además, se encarga de localizar las fallas y aislarlas. Sin embargo, su desventaja principal es que este tipo de configuración de red puede sufrir de saturaciones y problemas en caso de que ocurriera una falla en el nodo central. Esta topología se observa en la figura 2.7.

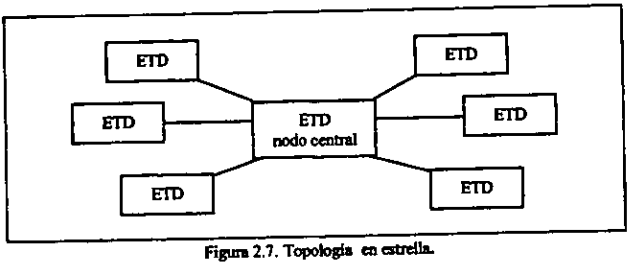


Figura 2.7. Topología en estrella.

2.3.5. Topología en anillo.

La topología en anillo se llama así por el aspecto circular del flujo de datos. En la mayoría de los casos, los datos fluyen en una sola dirección, y cada estación recibe la señal y la retransmite a la siguiente del anillo. Una ventaja que presenta es que en ella resultan bastante raros los embotellamientos. Su funcionamiento es el siguiente, cada componente sólo ha de llevar a cabo una serie de tareas muy sencillas: aceptar los datos, enviarlos al equipo terminal conectado al anillo o retransmitirlos al próximo componente del mismo. La desventaja que presenta es que todos los componentes del anillo están unidos por un mismo canal y si falla el canal entre dos nodos, toda la red se interrumpe. Por eso algunos fabricantes han ideado diseños especiales que incluyen canales de seguridad, por si se produce la pérdida de algún canal. Otros fabricantes construyen conmutadores que redirigen los datos automáticamente, saltándose el nodo averiado hasta el siguiente nodo del anillo con el fin de evitar que el fallo afecte a toda la red (Figura 2.8).

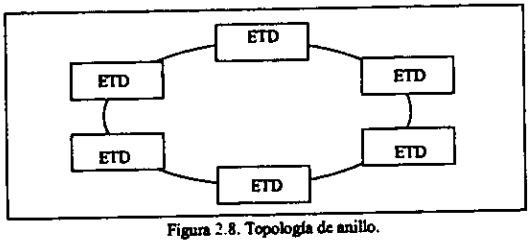


Figura 2.8. Topología de anillo.

2.3.6. Topología en malla.

Se podría decir que este tipo de configuración de red sería la ideal por su inmunidad a los problemas de embotellamiento y averías. Está constituida por múltiples caminos que ofrecen los equipos terminales y los de comunicaciones y es posible orientar el tráfico por trayectorias alternativas en caso de que algún nodo esté averiado u ocupado. La gran desventaja es que su realización requiere de un método complejo y costoso. Ver figura 2.9.

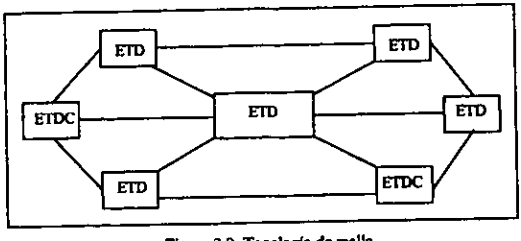


Figura 2.9. Topología de malla.

2.4 Clasificación de las redes según su tecnología de transmisión y su área geográfica.

No existe una clasificación general donde se consideren todos los diferentes tipos de redes de computadoras, sin embargo, sobresalen dos características a partir de las cuales podrían clasificarse en tecnología de transmisión y el área geográfica:

- Cuando se habla de tecnología de transmisión se hace referencia a la forma en que se envían los datos, la cual puede ser ***punto a punto*** o por ***difusión.***

  Una red punto a punto tiene la característica de realizar múltiples conexiones entre máquinas y cuando un paquete es enviado pasa por uno o más nodos antes de llegar a su destino. En este tipo de red existen diferentes caminos para poder llegar a un punto, por lo que se emplean los algoritmos de ruteo para escoger la ruta óptima.

  Las redes de difusión tienen un solo canal de comunicación compartido por todas las máquinas de la red. Los paquetes o mensajes cortos que envía una máquina son recibidos por todas las demás pero, sólo es aceptado por aquella que coincida con el campo de dirección especificado dentro del paquete.

  Los sistemas de difusión generalmente ofrecen la posibilidad de dirigir un paquete a todos los destinos, colocando un código especial en el campo de dirección. Cuando se transmite un paquete con este código, cada máquina en la red lo recibe y lo procesa. Este modo de operación se llama difusión (broadcasting). Cuando los sistemas de difusión realizan la transmisión a un subconjunto de máquinas se dice que están operando en multidifusión. Para llevar a cabo esta transmisión, se reserva en el paquete o paquetes a enviar un bit para indicar multidifusión. Los restantes n-1 bits de dirección pueden contener un número de grupo. Cada máquina puede pertenecer a un grupo o a todos, de tal forma que cuando se envíe un paquete a cierto grupo, se entrega a todas las máquinas pertenecientes a ese grupo.

  Las redes de difusión se pueden dividir también en estáticas y dinámicas, dependiendo de como se asigna el canal. Una asignación estática, divide el tiempo en intervalos discretos y ejecuta un algoritmo de asignación cíclica, permitiendo a cada máquina transmitir únicamente cuando le llega su turno. La asignación estática desperdicia la capacidad del canal cuando una máquina no tiene nada que decir durante su segmento asignado, por lo que muchos sistemas intentan asignar el canal dinámicamente (es decir, por demanda).

  Los métodos de asignación dinámica para un canal común son centralizados o descentralizados. En el método de asignación de canal centralizado hay una sola entidad, por ejemplo una unidad de arbitraje que determina quien es el siguiente. En el método de asignación de canal descentralizado no hay una entidad central; cada máquina debe decidir por si misma si transmite o no.

- El otro criterio para clasificar las redes es respecto a su área geográfica y en este sentido, dependiendo de la extensión física que cubra la red, ésta puede ser una red local, metropolitana , de área amplia o global.

2.4.1. Red de área local.

Las redes de área local, llamadas LAN (Local Area Network), son redes de propiedad privada dentro de un solo edificio o campus de hasta unos cuantos kilómetros de extensión. Se usan ampliamente para conectar computadoras personales y estaciones de trabajo en oficinas de compañías y fábricas con el objeto de compartir recursos e intercambiar información. Las LAN se distinguen por tres características: el tamaño, la tecnología de transmisión y la topología.

Este tipo de redes están restringidas en tamaño, lo cual significa que el tiempo de transmisión del peor caso está limitado y se conoce de antemano. Otra característica de una red local es la tasa de errores que suele ser considerablemente menor que la de las redes extensas. Conocer los límites y ventajas hace posible usar ciertos tipos de diseños que de otra manera no serían prácticos, y también simplifica la administración de la red.

Las redes de área local usan una tecnología de transmisión que consiste en un cable sencillo al cual están conectadas todas las máquinas, tradicionalmente operan a velocidades de 10 a 100 Mbps, pero en la actualidad, con el surgimiento de ATM (Asincronous Transfer Media) podemos hablar de velocidades del orden de megabits. (Figura 2.10)

Existen diversas topologías que pueden implantarse, por ejemplo, la IEEE 802.3 llamada Ethernet es una red de transmsión cuya topología empleada es la de bus con control de operación descentralizado a 10 o 100 Mbps. Las computadoras de una Ethernet pueden transmitir cuando quieran; si dos o más paquetes chocan, cada computadora sólo espera un tiempo al azar y lo vuelve a intentar, este mecanismo de arbitraje se le conoce como CSMA/CD (Carrier Sense Multiple Acces /Collision Detection). Otro segundo tipo de sistema de difusión es la IEEE 802.5 (el token ring de IBM) basada en anillo que opera a 4 y 16 Mbps.

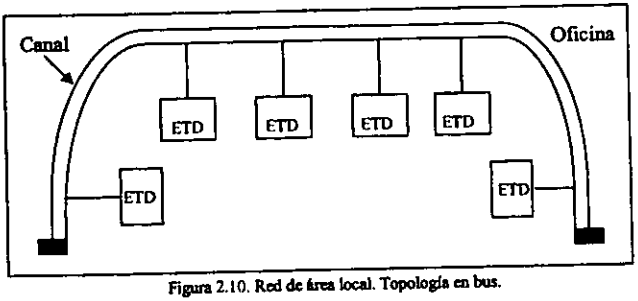


Figura 2.10. Red de área local. Topología en bus.

2.4.2. Red de área metropolitana.

Una red de área metropolitana, o MAN (metropolitan area network) es básicamente una versión más grande de una LAN. Su extensión puede abarcar un grupo de oficinas coorporativas cercanas o una ciudad e incluso puede ser **privada o pública**. Una MAN sólo tiene uno o dos cables y no contiene elementos de conmutación, simplificando su diseño.

La principal razón para distinguir las MAN como una categoría especial es que se ha adoptado un estándar para ellas llamado DQDB (Distributed Queue Dual Bus o bus dual de cola distribuida, 802.6 es el número de la norma IEEE que lo define). El DQDB consiste en dos buses (cables) unidireccionales, a los cuales están conectadas todas las computadoras (Figura 2.11). Cada bus tiene una cabeza terminal (head-end), un dispositivo que inicia la actividad de transmisión. El tráfico destinado a una computadora situada a la derecha del emisor usa el bus superior. El tráfico hacia la izquierda usa el de abajo.

Un aspecto clave de las MAN es que hay un medio de difusión (dos cables, en el caso de la 802.6) al cual se conectan todas las computadoras.

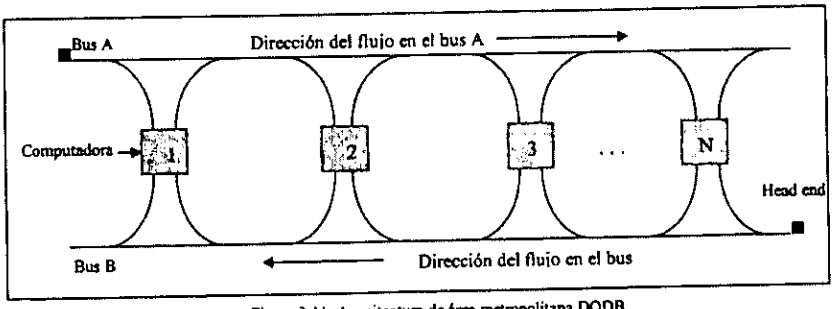


Figura 2.11. Arquitectura de área metropolitana DQDB

Características adicionales:

- La MAN puede proporcionar servicios de ***conmutación de paquetes*** y ***conmutación de circuitos.***
- La MAN puede proporcionar un servicio no orientado a la conexión.
- El nivel DQDB y el nivel físico, son independientes el uno del otro de modo que se pueden usar diversos niveles físicos. Por ejemplo SONET a 155 Mbps se apoya sobre el medio físico.

2.4.3. Red de área amplia.

Una red de área amplia o WAN (wide area network), constituye un sistema de comunicación que interconecta sistemas de computadoras geográficamente remotos. Las computadoras enlazadas son máquinas dedicadas a ejecutar programas de aplicación llamadas **host** (computadora central) y la comunicación que realizan fuera de las propiedades de una organización (edificios o campus) es a través de la red pública de teléfono, pero una organización podría crear sus propios enlaces WAN mediante microondas, satelitales u otras tecnologías de la comunicación.

Las redes de área amplia suelen estar constituidas por líneas de transmisión y elementos de conmutación. Las líneas de transmisión (también llamadas circuitos, canales o troncales) mueven bits de una máquina a otra. Los elementos de conmutación son computadoras especializadas que conectan dos o más líneas de transmisión. Cuando los datos llegan por una línea de entrada, el elemento de conmutación debe escoger una línea de salida para reenviarlos, a estas computadoras se les denomina nodos conmutadores de paquetes, sistemas intermedios, centrales de conmutación de datos o ***ruteadores***.

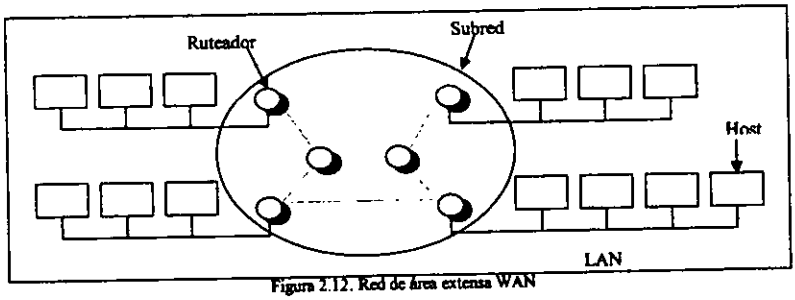


Figura 2.12. Red de área extensa WAN

Como se observa en la figura 2.12 la colección de líneas de comunicación y ruteadores forman la subred. Cuando se envía un paquete de un ruteador a otro a través de uno o más ruteadores intermedios, el paquete se recibe completo en cada ruteador intermedio, se almacena hasta que la línea de salida requerida está libre y a continuación se reenvía. Una subred basada en este principio, se llama de punto a punto, de almacenar y reenviar, o de paquete conmutado. Cuando los paquetes son pequeños y el tamaño de todos es el mismo, suelen llamarse celdas.

Cuando se usa una subred punto a punto, una consideración de diseño importante es la topología de interconexión del ruteador. Las redes de área amplia típicamente tienen topologías irregulares.

Una segunda posibilidad para una WAN es un sistema de satélite o de radio en tierra. Cada ruteador tiene una antena por medio de la cual puede enviar y recibir. Todos los ruteadores pueden oír las salidas enviadas desde el satélite y en algunos casos pueden también oír la transmisión ascendente de los otros ruteadores hacia el satélite. Algunas veces los ruteadores están conectados a una subred punto a punto de gran tamaño, y únicamente algunos de ellos tienen una antena de satélite.

2.4.4. Red Global.

Una red global (GAN) es una colección de redes interconectadas. Cada red es diferente una de otra, en cuanto a su tamaño, tecnología de transmisión y topología, por lo que lograr su comunicación requiere de ciertas máquinas cuyo propósito es el de realizar las traducciones necesarias en términos de hardware y software para lograr la conexión. Estas máquinas son conocidas como ***pasarelas*** (gateways).

Las conexiones que existen en esta red incluyen a la red de enlace telefónico, a los enlaces por microondas terrestres o satelitales y redes de fibra óptica.

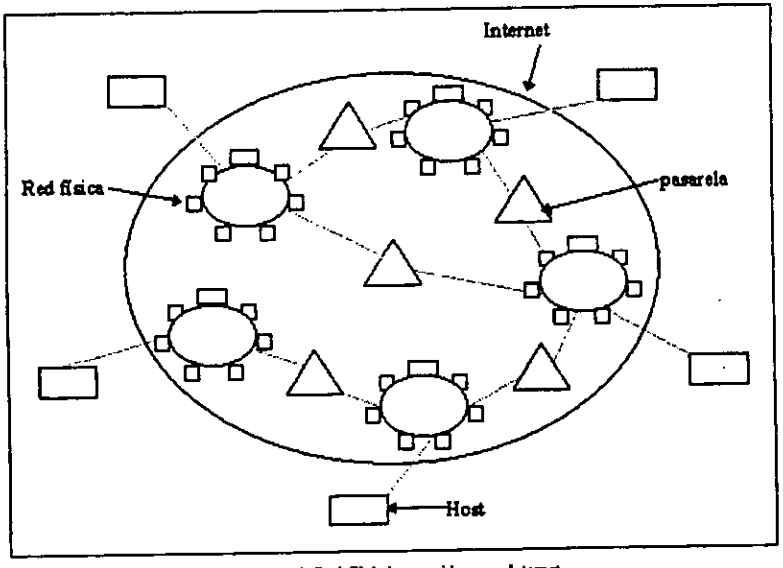


Figura 2.13. Red Global, conocida como Internet.

Como se observa en la figura 2.13, la red global más conocida es Internet (red de redes), la cual conecta redes ubicadas en escuelas, bibliotecas, hospitales, oficinas de gobierno, compañías y otras. Es una estructura con vínculos a muchas redes públicas y privadas. El acceso a estas redes puede ser libre, y sin restricciones, en función de los privilegios de acceso o de cuanto se este dispuesto a gastar.

Adicionalmente podemos señalar que Internet proporciona vastas cantidades de información útil y oportuna para estas instituciones mediante los enlaces de telecomunicación existentes.

2.5. Redes inalámbricas.

Mediante la radiocomunicación es posible la existencia de redes inalámbricas y de la informática móvil. Una red inalámbrica es aquella que no requiere cables para la interconexión de las estaciones de trabajo, pero necesita instalar en una oficina central un ***transceptor*** encargado de difundir las señales. En una LAN inalámbrica se requiere de un transceptor, que se conecta mediante un cable a un ETD o a otro segmento de la red.

Existen dos tecnologías que compiten en la radiocomunicación móvil, una basada en técnicas de radio y enlaces satelitales y la otra en el uso de la telefonía celular.

Una de las facilidades que brinda esta tecnología a una persona que realiza constantes viajes es la de poder conectarse a su base de operaciones y realizar su trabajo como si estuviera en su oficina o en su casa. El uso del equipo electrónico portátil se emplea para enviar y recibir llamadas telefónicas, faxes, correo electrónico, leer archivos remotos, entrar en máquinas remotas etc.; además, es posible hacer esto desde cualquier lugar ya sea en tierra, mar o aire.

Las redes inalámbricas son de gran valor para que los camiones, taxis, autobuses, y las personas que hacen reparaciones, mantengan contacto con su base. También pueden usarlas los rescatistas en sitios de desastre (incendios, inundaciones, temblores, etc.), donde se ha dañado el sistema telefónico.

2.6. La red telefónica.

Los equipos terminal de datos, de comunicaciones y de conmutación están conectados entre sí a través de un canal telefónico. Los clientes o usuarios, ya sean domésticos u oficinas, se conectan al sistema telefónico a través de una central llamada local o final. El enlace con esta central se lleva por medio de un cable de pares llamado lazo local o lazo de abonado. Ver figura 2.14.

Si un suscriptor conectado a una central local determinada llama a otro suscriptor conectado a la misma central local, el mecanismo de conmutación dentro de la oficina establece una conexión eléctrica directa entre los dos lazos locales. Esta conexión permanece intacta mientras dura la llamada.

Si el equipo terminal al que se llama está conectado a otra central local, se tiene que usar un procedimiento diferente. Cada central local tiene varias líneas salientes a uno o más centros de conmutación cercanos llamados centrales de cargo (si están dentro de la misma área local son centrales tándem). Estas líneas se llaman troncales de conexión con cargo. Si sucede que tanto la central final de quien llama como la de quien es llamado tienen una troncal de conexión a la misma central de cargo, la conexión se puede establecer dentro de la oficina de cargo.

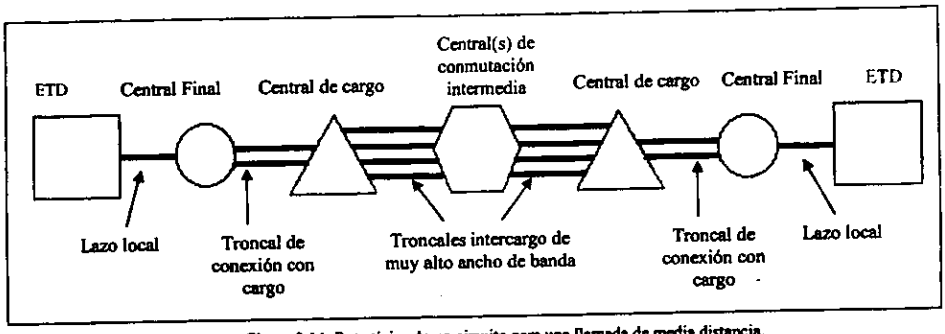


Figura 2.14. Ruta típica de un circuito para una llamada de media distancia.

Si el equipo terminal que llama o el llamado no tienen una central de cargo común, la trayectoria se deberá establecer en un nivel más alto de la jerarquía. Hay centrales primarias, seccionales y regionales que forman una red que conecta a las centrales de cargo. Las centrales de cargo primarias, seccionales y regionales se comunican entre sí mediante troncales intercargo de gran ancho de banda (llamadas también troncales interoficinas).

Se trata de encaminar la llamada por la trayectoria más económica, que suele coincidir con el camino más corto y/o con el menor número posible de conmutaciones. Este método reduce el retardo de establecimiento del enlace con el otro equipo terminal de datos y el menor número de conmutaciones intermedias disminuye el costo para la compañía telefónica. A medida que aumenta el recorrido por las centrales tándem, aumenta el número de componentes implicados, lo cual se traduce en un retardo mayor y un costo adicional.

El sistema está construido en torno a diversas líneas de alta capacidad que transportan la mayor parte del tráfico. Estas líneas se instalan cuando el volumen de llamadas justifica el establecimiento de canales de gran capacidad entre dos centrales. Por tanto, la configuración concreta de estas líneas dependerá del volumen del tráfico que exista entre las dos centrales. En un principio el sistema intenta conmutar la llamada hacia algún nivel inferior de la jerarquía, a través de la jerarquía, o como último recurso hacia arriba de la misma. El hecho de enviar la llamada a través de niveles superiores de la jerarquía suele implicar más conmutaciones intermedias, lo cual aumenta el retardo de conexión y eleva el costo del enlace de la compañía.

2.7. Conexión entre el mundo analógico y el digital.

En nuestros días, son muchos los componentes de los equipos de transmisión que emplean tecnología digital, debido a que ofrecen las siguientes ventajas sobre los sistemas analógicos:

- Regenerativos. Aunque la atenuación y la distorsión también existen en una señal digital es más fácil regenerarla, debido a que la forma de onda está representada por niveles de tensión discretos. Las señales

digitales pueden reconstruirse totalmente antes de que se deteriore detectando la ausencia o presencia de un pulso en lugar de captar la amplitud de una señal analógica.

- Multiplexaje. Se pueden intercalar voz, datos, música e imágenes para aprovechar de forma más eficiente los circuitos y el equipo.
- Detección de errores. Un bit transmitido se recibe bien o no, con lo que se simplifica el rastreo de problemas.
- Costo menor. La transmisión digital es mucho más económica que la analógica, puesto que no es necesario reproducir exactamente una forma de onda analógica; basta poder distinguir correctamente un 0 y un 1.

Actualmente, las troncales de larga distancia dentro del sistema telefónico se están convirtiendo rápidamente a tecnología digital. El sistema antiguo utilizaba transmisión analógica por cables de cobre; el nuevo emplea transmisión digital por fibra óptica.

Aunque los troncales de larga distancia se están convirtiendo en digitales en algunos países, los lazos locales todavía son analógicos y en consecuencia cuando una computadora o terminal desea enviar datos digitales, los datos se deben convertir primero en forma analógica para ser transmitidos por el lazo local, luego convertirse en forma digital para transmitirse por los troncales de largo alcance, después convertirse a analógicos en el lazo local del extremo receptor y por último, a digitales para almacenarse en la computadora destino.

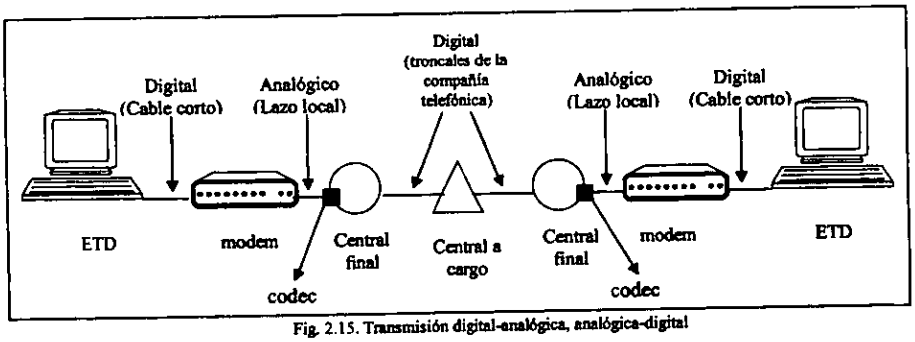


Fig. 2.15. Transmisión digital-analógica, analógica-digital

En el esquema de la figura 2.15, observamos que el dispositivo que realiza la conversión de la señal digital a analógica es el equipo de comunicación de datos llamado modem. Este dispositivo permite que un ETD digital transmita datos a otro ETD receptor a través de un canal analógico. La palabra modem es una abreviatura de modulador/demodulador. El proceso consiste en modular la señal en el modem emisor y demodularla en el modem receptor. Por otra parte, el dispositivo que realiza la conversión de la señal analógica a digital se llama ***codec***.

La definición precisa de modulación es la siguiente: modificación de una señal periódica para transportar datos. Esta señal periódica es lo que se conoce como portadora. Los datos que modulan la portadora (es decir, los datos que proceden del ETD), constituyen la señal en banda base.

El modem se encarga de modificar la señal portadora (ya sea en su amplitud, en su frecuencia o en su fase) para poder transportar la señal en banda base. Como se muestra en la figura 2.16.b, un modem en ASK modifica la

amplitud de su portadora de acuerdo con el flujo de bits que ha de enviar. En este caso, una amplitud más elevada representa un cero, y una amplitud más baja representa un uno. Un sistema más extendido es la FSK (modulación en frecuencia), que consiste en variar la frecuencia, manteniendo constante la amplitud. En la figura 2.16.c un 1 binario se representa con una determinada frecuencia, y un 0 binario con otra distinta. Otro sistema es la modulación PSK (modulación en fase), que consiste en alterar la fase para representar el cambio de 1 a 0 ó de 0 a 1. Observe la figura 2.16.d.

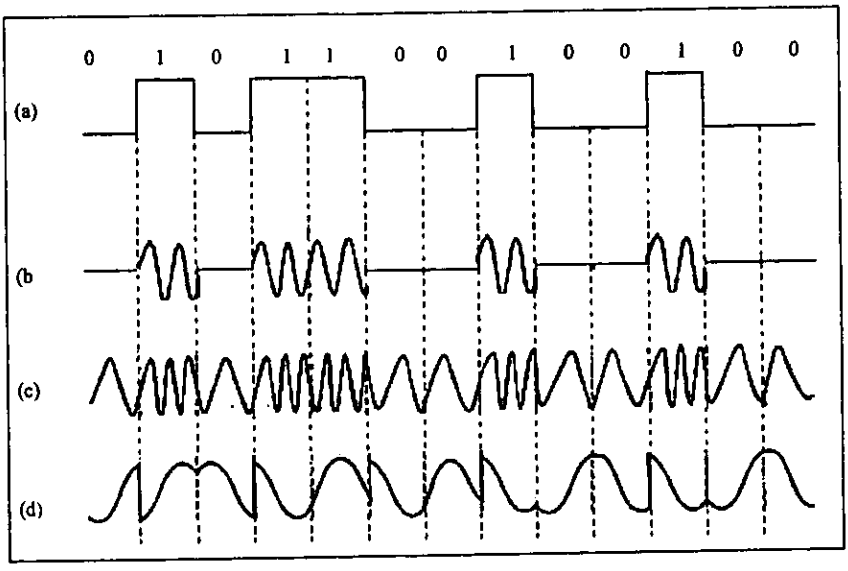


Figura 2.16. (a) Señal binaria. (b) Modulación de amplitud. (c) Modulación de frecuencia. (d)Modulación de fase.

En el caso de las líneas rentadas, es posible trabajar en forma digital desde el principio hasta el fin, sin embargo, éstas suelen ser muy caras y sólo se podrían considerar para la construcción de redes privadas.

2.8. Sincronización y autosincronización.

Una vez establecida la comunicación entre los ETD se necesita un método con el que ambos dispositivos lleven el control de la transmisión en curso. El transmisor debe enviar su señal de modo que el dispositivo receptor sepa cuando buscarla y reconozca cada 1 y cada 0 que vaya llegando. Para esto debe de existir un reloj en común entre los dispositivos que emiten y los que reciben, de tal forma que informe sobre el envío de los datos. Si el emisor se limita a enviar los datos por el canal sin previo aviso, lo más probable es que el receptor no tenga tiempo suficiente para ajustarse al flujo de datos que empieza a llegarle, en cuyo caso los primeros bits de la transmisión se perderán. Este proceso forma parte de un protocolo de comunicación, y suele conocerse como sincronización. Las conexiones de corta distancia entre máquinas suelen servirse de un canal aparte, o de una línea, para proporcionar la

sincronización. Esta línea transmite una señal que se activa y desactiva de acuerdo con determinadas normas preestablecidas. Cuando esta señal de reloj que llega por la línea cambia de estado, indica al dispositivo receptor que debe examinar la línea de datos. Asimismo, puede sincronizar el reloj del receptor de manera que éste quede alineado con total exactitud con cada bit que vaya entrando. Las señales de sincronismo temporización desempeñan dos funciones importantes: sincronizan al receptor con la transmisión antes de que lleguen los datos propiamente dichos y mantienen al receptor sincronizado con los datos que van llegando.

El inconveniente que presenta la señal de sincronismo es que puede verse adelantada o retardada en relación con la señal de datos, lo cual puede provocar que el receptor tenga dificultad de recibir el flujo de datos adecuadamente. Por otra parte, cuando las distancias entre los ordenadores y terminales son grandes, resulta más económico incorporar la temporización a la propia señal que usar un canal de sincronismo aparte. Para resolver los problemas anteriores se pueden emplear los ***códigos de autosincronizado***. Un código autosincronizado es aquél que permite al receptor comprobar periódicamente si está muestreando la línea en el momento exacto en que llega un bit de datos. Los mejores códigos autosincronizados son aquellos en los cuales el estado de la línea cambia muy frecuentemente, ya que estos cambios de estado permiten al receptor seguir reajustando su propio funcionamiento de acuerdo con la señal. La idea consiste en disponer de un código que presente transiciones regulares y frecuentes sobre el canal. Las transiciones se limitarán al tamaño de las divisiones correspondientes a los datos binarios existentes en el receptor; la lógica de muestreo consiste en buscar constantemente las transiciones de estado para delimitar los bits que vayan llegando.

El receptor suele realizar el muestreo a una velocidad mayor que la de llegada de los datos, para poder definir con mayor precisión el tamaño de los intervalos de cada bit. Algunos ejemplos de códigos autosincronizados son:

1. *Código sin retorno a cero* (NRZ). Se emplea en los sistemas de comunicaciones.
2. *Código de retorno a cero* (RZ). Se emplea en sistemas asociados a redes locales, fibras ópticas y tecnologías relacionadas con la luz.
3. *Código de Manchester*. Este código se utiliza mucho en grabaciones de cinta magnética, enlace de fibra óptica, líneas coaxiales y redes de área local.
4. *Código AMI bipolar*. Su estructura de señalización es empleada en el código PCM utilizado por empresas como AT&T y otras compañías telefónicas.

2.9. Red Pública.

Una organización tiene tres alternativas cuando construye redes sobre áreas extensas, las cuales pueden ser: utilizar redes privadas, redes públicas o una combinación de ambas.

Una red pública de datos (PDN, Public Data Network) es una red nacional de puntos de acceso y conmutadores interconectados que proporcionan transmisión simultánea de datos para muchos usuarios entre muchos puntos.

Hay dos tipos principales de PDN: la PDN de conmutación de circuitos y la PDN de conmutación de paquetes.

Las redes de conmutación de circuitos proporcionan un modo para establecer un canal físico de comunicación a través de la red pública entre el equipo que llama y el equipo llamado. La red telefónica utiliza la conmutación de circuitos para la comunicación entre los distintos ETD (computadoras y terminales), y las características que la distiguen son:

- Una vez establecida una llamada, los usuarios disponen de un enlace directo a través de los distintos segmentos de la red.
- Los conmutadores no poseen medios de almacenamiento intermedio.
- Debido a la ausencia de medios de almacenamiento, un conmutador puede quedar bloqueado (es lo que sucede cuando una llamada comunica).

La conmutación de circuitos es la necesidad de establecer una trayectoria de un extremo a otro antes de que se pueda enviar cualquier dato. El tiempo que transcurre entre que se termina de marcar y que el timbre comienza a sonar puede ser de 10 seg. y más en las llamadas de larga distancia. Durante este intervalo de tiempo, el sistema telefónico está buscando una trayectoria de cobre. En la figura 2.17 se muestra un esquema de la conmutación de circuito pero antes de que pueda comenzar la transmisión, la señal de petición de llamada se debe propagar hasta el destino, y debe ser reconocida. Una vez establecida la trayectoria, el único retardo de los datos que existe es el tiempo de propagación de la señal electromagnética que se encuentra alrededor de 5 mseg. por cada 1000 km. Por otro lado, se puede decir que la ventaja que ofrece este tipo de conmutación es que no hay peligro de congestión en la trayectoria establecida; esto es, una vez que la llamada entró no hay posibilidad de obtener una señal de ocupado, aunque podría obtener una antes de establecer la conexión, debido a la falta de capacidad de conmutación o de troncal.

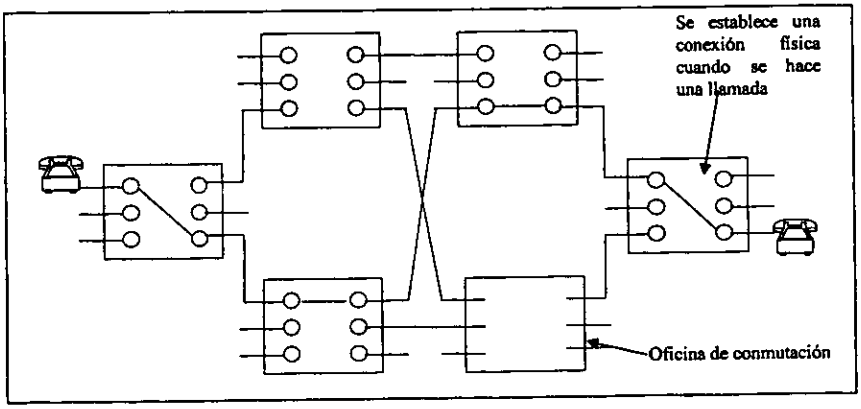


Figura 2.17. Conmutación de circuitos.

Una estrategia de conmutación alterna es la conmutación de mensajes. A diferencia de la conmutación de circuitos, la conmutación de mensajes es una tecnología que permite grabar la información para atenderla después, esto es debido

a la capacidad de almacenamiento (discos) que posee el conmutador (ruteador). Primero la computadora examina la dirección que aparece en la cabecera del mensaje y encamina el paquete hacia el ETD que ha de recibirlo. El mensaje (marco) se almacena en el ruteador y el de mayor prioridad permanece menos tiempo en cola que el de prioridad más baja. Cada marco se recibe en su totalidad, se inspecciona en busca de errores y después se retransmite. A esto se le conoce como una red de "almacenar y reenviar".

Con la conmutación de mensajes, no hay límite para el tamaño de los marcos, éstos pueden ser tan largos como se deseen, pero también significa que un solo marco puede acaparar una línea de ruteador a ruteador durante minutos, lo que hace inútil la conmutación de mensajes para el tráfico interactivo. Con el fin de resolver este problema se inventó la conmutación de paquetes. Las redes de conmutación de paquetes establecen un límite superior al tamaño del marco, lo que permite almacenar los paquetes en la memoria principal del ruteador en lugar de hacerlo en disco. Sí el tamaño de los paquetes se mantiene minimizado, cuando ocurra un fallo en la red y se alteren los datos en tránsito, sólo los paquetes afectados necesitarán retransmitirse en lugar de realizar una retransmisión completa.

Los ruteadores en la red dirigen los paquetes a su destino, ofreciendo servicios sin conexión. Un servicio no orientado a la conexión no establece un canal de comunicación previo dedicado para la transmisión de datos. Aunque las redes de conmutación de paquetes proporcionan servicio sin conexión entre dos puntos cualesquiera, es posible establecer un ***circuito lógico*** (circuito virtual) a través de una red de conmutación de paquetes. En los circuitos lógicos los paquetes fluyen en una forma ordenada desde el origen al destino y se elimina el retardo asociado con el reemsamblado de paquetes en el destino. Observe la figura 2.18.

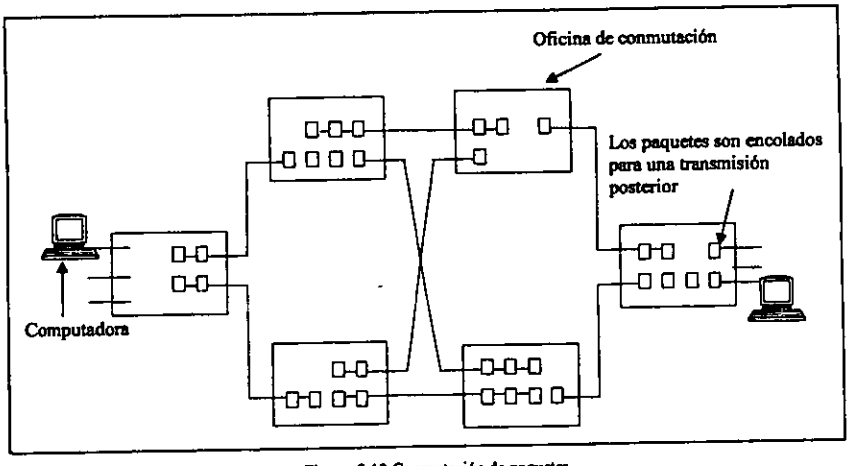


Figura. 2.18.Conmutación de paquetes.

El servicio de conmutación de paquetes tradicional ha sido ***X.25***, pero existen otros servicios que permiten la construcción de conexiones de red rápidas sobre facilidades de conmutación públicas como son: ***Frame Relay***, Servicio de conmutación de datos multimegabit (***SMDS***, Switched Multimegabit Data Service), y el modo de transferencia asíncrono (***ATM***, Asynchronous Transfer Mode). Todos estos servicios requieren una línea especial

desde el lugar del cliente al punto de acceso a la red de conmutación de paquetes. Los servicios están disponibles en tarifas de líneas alquiladas o en tarifas de líneas de enlace telefónico. Una red PDN típica, forma una malla de conexiones mundiales entre el equipo de conmutación PDN. Los circuitos constan de sus propias ***líneas privadas*** o de líneas PDN alquiladas en las principales compañías de telecomunicaciones.
De lo anterior se puede decir que una red pública PDN permite entre otras cosas:

- Consolidar la red privada, permitiendo compartir información con todas sus sucursales.
- Integrar las actividades de una empresa que tiene sucursales en diferentes lugares geográficos.
- Extender la cobertura nacional e internacional de cada empresa cuando ésta lo desee.
- Tener acceso a la información de las operaciones que se efectúen dentro de la organización. Contando con seguridad interna y permisos para cada uno de los usuarios pertenecientes a dicha empresa.

En la tabla 2.1, se presentan algunas empresas que ofrecen servicios de redes públicas, así como también la tecnología que emplean, los servicios, la cobertura y los proveedores de información conocidos como carriers.

Red Pública	Tecnología	Servicios	Cobertura geográfica	Conectividad (carrier)
Alestra	Acceso de nodos en México, a través de la infraestructura y plataforma de la red de AT&T.	X.25, Frame Relay, ATM, Internet, videoconferencia.	Mundial, a través de su red inteligente.	A través de Alestra, AT&T y en algunos casos Telmex.
Avantel	Tres conmutadores centrales y dos internacionales, basados en fibra óptica.	X.25 y Frame Relay, videoconferencia	Nacional e Internacional	A través de Telmex
Global One	Telecomunicaciones a nivel global basadas en una infraestructura digital	X.25, Frame Relay, Internet, videoconferencia	Nacional e Internacional	A través de Telmex
GES	Enlaces dedicados basados en fibra óptica	Frame Relay y TCP/IP	Nacional e Internacional	A través de MCI, AT&T y en América Latina se apoyan del carrier más fuerte de cada país
Infosat	Varios transpondedores de banda C de la región 2 de los satélites Solidaridad, basados en una topología estrella punto-multipunto.	Canal dedicado multiprotocolo, ATM, Frame Relay, videoconferencia.	Nacional, norte de EU, algunos países de Latinoamérica y el Caribe	A través de RedSat en la mayoría de los casos
Unicom	Frecuencias satelitales basadas en líneas de interconexión y tres conmutadores principales unidos con fibra óptica.	X.25, Frame Relay, ATM, Internet.	Nacional	A través de GTE y Telefónica de España.
UniNet	Varios conmutadores montados sobre una red de fibra óptica SDH de alta capacidad	Enlace multiprotocolo nacional e internacional basado en X.25, Frame Relay, ATM y TCP/IP	Nacional e Internacional	A través de Telmex

Tabla. 2.1. Algunas redes públicas.

2.10. Red Privada.

Una red privada es una red construida con facilidades públicas, excepto que el cliente proporciona todo el equipo de conmutación y establece líneas alquiladas entre distintos lugares. Privada se refiere al hecho de que la organización tiene uso completo de las líneas y no las comparte con otros. Otra manera de construir redes privadas es usar instalaciones de microondas o servicios de red de área metropolitana de distintos proveedores de LECs (Local

Exchange Carrier). La mayoría de las compañías utilizan los servicios proporcionados por las redes públicas de datos para las redes de área extensa pero construyen facilidades de red privadas para la conexión de facilidades locales.
La red privada consiste en un equipo de conmutación y de comunicación que es propiedad de una organización interconectada mediante líneas de comunicación alquiladas o propias. Las líneas privadas facilitan a la compañía el mantenimiento de la seguridad y el control sobre el tráfico que atraviesa la línea.
Una línea alquilada es un circuito de comunicación que se establece de forma permanente por la compañía telefónica, evitando entrar en contacto con el equipo de conmutación en la compañía de telecomunicaciones de intercambio local (LEC), y por tanto no necesitan una fase de establecimiento antes de cada transmisión de datos. Si la línea es de larga distancia se implica a una compañía de telecomunicación de larga distancia, así como a la LEC en el otro extremo del circuito. La línea abarca el lazo local, que es el cable que conecta un determinado negocio con la LEC, y se acondiciona este cable para asegurar una cierta calidad, como el uso de velocidades más altas de transferencia de datos en la línea.
Las líneas alquiladas pueden consistir en circuitos analógicos o digitales:

- Las líneas analógicas requieren modems en cada extremo y típicamente proporcionan las mismas velocidades de datos que la línea de enlace telefónico, excepto en aquellas situaciones en las que el cliente las contrata con la compañía telefónica para mantener la línea disponible para uso inmediato cuando sea necesario.
- Los circuitos digitales son líneas acondicionadas que pueden proporcionar velocidades de transmisión tan altas como las líneas analógicas, hasta 45 Mbps (para líneas T3).

Las redes privadas construidas con líneas alquiladas T1(1.544 Mbps) o T3 (45 Mbps) son adecuadas en algunas condiciones, en función de los presupuestos, los requisitos y la distancia entre los puestos tal y como se describe a continuación:

- El costo de las líneas alquiladas se incrementa con la distancia, así que sólo son apropiadas dentro de ciertas áreas geográficas.
- El servicio de línea digital normalizado es el canal T1, que proporciona velocidades de transmisión de 1.544 Mbps.
- Las líneas T1 pueden transportar voz y datos mediante el uso de dispositivos multiplexores, de modo que usan frecuentemente para proporcionar conexiones telefónicas entre lugares remotos pertenecientes a una organización.
- Una línea T1 puede proporcionar 24 canales de voz o datos en un ancho de banda de 64 Kbps. Los clientes que no necesiten el ancho de banda completo de T1 pueden optar por T1 fraccional, que proporciona servicios digitales en incrementos de 64 Kbps. Por otra parte una línea T3 puede proporcionar el equivalente de 28 líneas T1 para usuarios que requieren mucho ancho de banda.

2.10.1. Centrales privadas de conmutación PBX.

El equipo de conmutación para las redes privadas también puede ser proporcionado por las centrales privadas de conmutación PBX conocidas como "centralitas". A principios de los ochenta, estas centrales se utilizaban fundamentalmente como dispositivos de conmutación telefónica para oficinas. Los teléfonos se conectaban mediante hilos a la centralita, la cual se encargaba de conmutar las llamadas entre las oficinas de la misma organización y los cables que conducían a la central telefónica principal. Las PBX han evolucionado hasta convertirse en potentes herramientas para la transmisión integrada de voz y datos para la configuración de redes.

Las centrales privadas de conmutación se conocen también como centrales privadas automatizadas de conmutación (PABX), o centrales de conmutación informatizada (CBX) o centralitas.

Las PBX más modernas han conseguido la conmutación completamente digital. En conmutación digital se manejan conceptos muy similares a los de la conmutación de paquetes, en cuanto no se dedica todo el ancho de banda de una conmutación a una conexión concreta, sino que varios usuarios pueden ahora compartir un mismo camino físico del conmutador, entrelazando las conexiones de voz o de datos durante los periodos de silencio en la conversación o de ausencia de transmisiones por parte del ETD. Las PBX actuales son procesadores de gran potencia capaces de realizar funciones como el encaminamiento óptimo. También ofrecen funciones como las repeticiones de marcado, la espera de llamada, o la expedición de llamada.

El sistema llamado RDSI Red Digital de Servicios Integrados (ISDN Integrated Services Digital Network), tiene como meta principal la integración de servicios de voz y sin voz. Una característica de RDSI son los teléfonos con múltiples botones para establecer llamadas inmediatas con teléfonos en cualquier parte del mundo. Otra posibilidad es un teléfono que exhibe el número, nombre y dirección de quien llama en una pantalla mientras el teléfono suena. Los servicios avanzados que no son de voz incluyen tomar la lectura del medidor de electricidad en forma remota y alarmas en líneas médicas, contra ladrones, y de humo que llaman en forma automática al hospital, a la policía o al departamento de bomberos, respectivamente, y proporcionan la dirección para agilizar la respuesta.

La idea clave en que se basa la RDSI es la del conducto digital de bits, un conducto conceptual entre el cliente y la portadora a través del cual fluyen los bits. En el siguiente esquema el dispositivo terminal de red NT1 está colocado en el local del cliente por el proveedor y lo conecta a la central RDSI en la oficina de la portadora, a varios kilómetros de distancia, usando el par trenzado. El dispositivo NT1 tiene un conector que puede insertar un cable de bus pasivo. Hasta ocho teléfonos, terminales, alarmas y otros dispositivos RDSI se pueden conectar al cable, de forma similar a como los dispositivos se conectan a una LAN. Sin embargo, en grandes empresas se manejan más conversaciones al mismo tiempo por lo que se necesita de otro dispositivo de conmutación el cual es un PBX. Como se muestra en la figura 2.19, el PBX es conectado al conmutador RDSI, el cual proporciona la interfaz real para teléfonos, terminales y otro equipo.

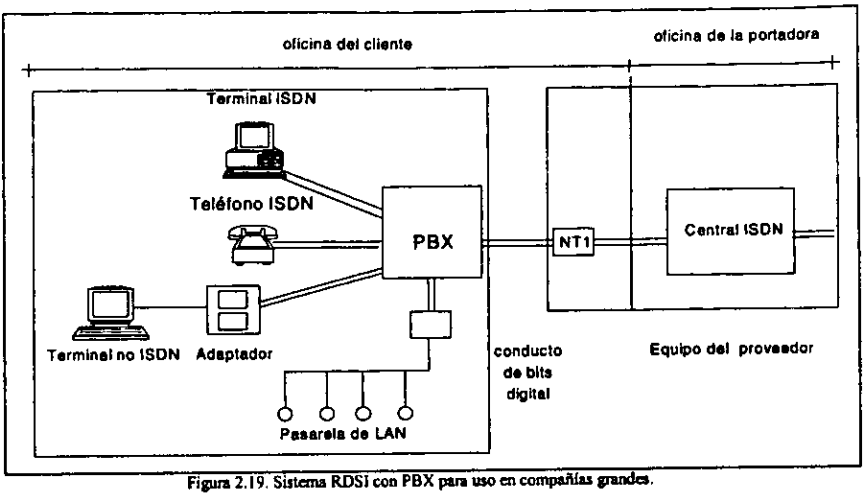


Figura 2.19. Sistema RDSI con PBX para uso en compañías grandes.

La mayoría de las empresas están incorporando servicios de redes públicas en sus redes privadas, ya que resulta menos costoso.

2.10.2. Intranet y Extranet.

Una Intranet es una red privada implementada por la propia empresa y suele constar de varias redes LAN (Ethernet conmutada, ATM, entre otras) que se interconectan mediante redes WAN tipo Frame Relay y ATM, líneas punto a punto, RDSI para acceso remoto, etcétera. Además, se caracteriza por funcionar justo como Internet al proporcionar servicios de tejido (web); los protocolos de comunicación TCP/IP y HTTP; y publicaciones en HTML. Sin embargo, estos servicios sólo se ofrecen a empleados selectos. Observe la figura 2.20.

Los usuarios remotos, estén donde estén, pueden comunicarse con la red corporativa a través de una llamada local al proveedor de Internet (ISP, Internet Service Provider). Se pueden conectar computadoras de distintas plataformas de hardware y de distintos sistemas operativos.

Para el montaje de la Intranet se requiere determinar y desarrollar la infraestructura necesaria, para lo cual se tiene que ver si se posee de una red basada en protocolo TCP/IP, ya que es el protocolo que se requiere para conectarse a Internet. En caso de que no exista, habrá que implementarlo. También se debe revisar si la infraestructura de red que tiene la empresa soportará el tráfico que se estima en la Intranet, y observar si el ancho de banda que se tiene es suficiente, ya que en una Intranet se manejan muchos archivos de gráficos, audio y video.

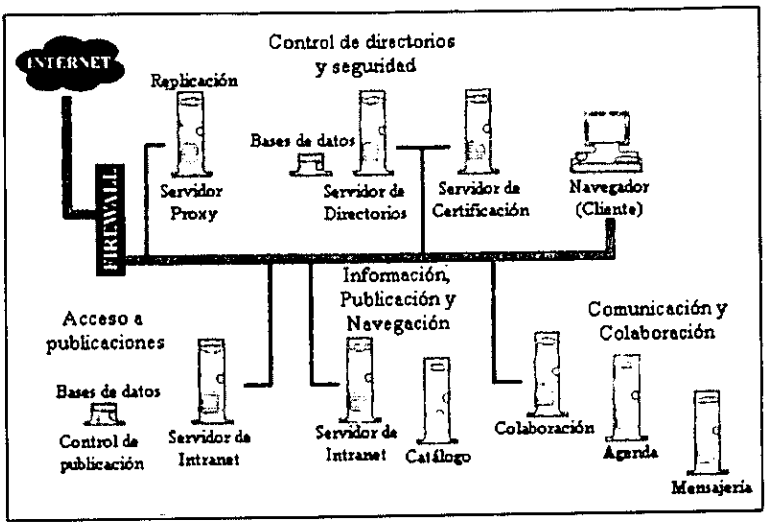


Figura. 2.20. Intranet.

Otro aspecto importante es decidir el tipo de conexión remota que se va a establecer, dependiendo de las necesidades de la organización. Algunos ejemplos son: las líneas dedicadas, las cuales ostentan velocidades de 56 Kbps a 1.54Mkbps; líneas no dedicadas de marcado, que pueden ser por marcado de un modem, o una red de servicios integrados digitales de 64 Kbps a 128 Kbps; otro tipo es el Frame Relay con velocidad de 56 Kbps a 1.54 Mbps; y la última de este tipo es ATM, que puede alcanzar velocidades a más de 45 Mbps.

Es importante determinar las necesidades de seguridad de la organización e implementarlas, especialmente para las organizaciones que implementan una Intranet, ya que con un sistema de seguridad se puede limitar el acceso a la información de la empresa solamente al personal autorizado. Algunas medidas de seguridad son los ***firewalls***, que limitan el acceso por medio de contraseñas otorgadas sólo al personal autorizado, y por medio de la encriptación, que es la codificación de los mensajes e información que va estar circulando por la Intranet.

La decisión de cuándo instalar una red privada o cuándo utilizar los servicios de una red pública depende esencialmente de las necesidades y recursos de la empresa. Dado que en muchos casos la elección dependerá del costo, las empresas de telecomunicaciones que ofrecen tanto las líneas privadas que se utilizan para construir las redes privadas como los servicios de redes públicas de conmutación de paquetes, ajustan sus tarifas para que cada mercado sea atractivo a diferentes aplicaciones. En la tabla 2.2 se realiza una comparación entre algunos aspectos relevantes de ambas redes para dar una idea de lo que pudieran ser los parámetros para tomar la decisión.

Muchas veces el resultado de escoger entre una red privada y una red pública de conmutación de paquetes puede ser una solución extrema por lo que las redes híbridas han ido ganando terreno dentro del mundo de las telecomunicaciones. Las redes híbridas son el resultado de utilizar los servicios de las redes públicas para intercomunicar redes privadas. Por ejemplo, dentro de una empresa la comunicación a nivel metropolitano puede

realizarse a través de una red privada y los accesos remotos a nivel nacional a través de una red pública, para reducir los costos de larga distancia. Otro caso se presenta cuando dentro de una empresa se comunica el corporativo con las oficinas regionales a través de una red privada, logrando tener las ventajas de la seguridad y administración, y para la comunicación con proveedores, clientes u otro tipo de acceso remoto se utilizan los servicios de las redes públicas, evitando elevar los costos de inversión.

	Redes Privadas	Redes Públicas
Enfoque	Interconexión y comunicación de oficinas de una misma empresa.	Comunicación fuera de las empresas. Interconexión de redes privadas.
Servicios	Cada servicio extra o implementación de nueva tecnología incurre en un costo de inversión.	Reducción en los costos de servicios de valor agregado. Facilidad para mantenerse con tecnología de vanguardia.
Seguridad	Se puede proporcionar la seguridad específica necesaria para la empresa.	El proveedor suele ofrecer cierto nivel de seguridad, pero por el hecho de ser una red pública el acceso de cualquier persona a la red es muy sencillo.
Control	Se tiene el control total sobre los recursos, administración y entrega de información.	El proveedor se encarga de la asignación de recursos dependiendo de los requerimientos de sus múltiples usuarios.
Inversión	Se requiere una inversión fuerte, no sólo de equipo sino de diseño, mantenimiento, operación y expansión.	Los costos en inversión se reducen dramáticamente, sólo se necesita invertir en equipo de acceso a la red.
Conectividad	Transmisión restringida a los nodos que conforman la red privada.	Capacidad de transmisión a todos los puntos de enlace de la red. Facilidad para dar de alta a nuevos usuarios.
Operación y Mantenimiento	Se necesita gente capacitada para mantener la operación eficiente de la red y dar el mantenimiento.	El proveedor de la red pública se encarga de todos los costos de operación y mantenimiento.

Tabla 2.2. Comparación entre redes privadas y redes públicas.

Sin embargo, las empresas se han dado cuenta de que sus colaboradores, socios e incluso sus clientes se quedaban fuera de sus Intranets, para poner remedio a esto surge la Extranet.

Una Extranet es una red externa de colaboración que utiliza la tecnología de Internet con el propósito de conectar a una empresa con sus proveedores, clientes u otros socios. El término Extranet describe el software que facilita la relación entre las distintas compañías y puede ser concebida como parte de una Intranet que es accesible para otras empresas, o como herramienta que permite la colaboración entre las mismas. Una red externa tiene las siguientes aplicaciones:

- Grupos privados que cooperan con la empresa y comparten la misma información e ideas.
- Entornos de colaboración, donde varias empresas participan en el desarrollo de una aplicación nueva que ellos pueden usar.
- Programas de formación u otros contenidos educativos que las empresas deseen desarrollar o compartir.
- Lista de catálogos de productos.
- Gestión de proyecto y control para empresas que forman parte de un mismo proyecto.

La información compartida es accesible sólo para aquellos miembros colaboradores de la empresa que posee la Intranet, aunque en algunos casos puede ser pública. Debido a estas características la Extranet requiere de un alto grado de seguridad y privacidad de la información; ésta podría obtenerse asegurándose que las líneas de transmisión fueran privadas o alquiladas o utilizando técnicas de seguridad a través de Internet.

2.10.3. Redes Virtuales Privadas.

Actualmente se está ofreciendo a las empresas una tecnología que permite un considerable ahorro en los costos de comunicaciones; soporte de usuarios remotos e infraestuctura; y favorecerá en gran medida un mejor aprovechamiento del equipo informático. Esta tecnología, llamada ***VPN*** (Virtual Private Network) o Red Privada Virtual, es una extensión de una red interna (Intranet) a través de una red pública como Internet que proporciona las características y los beneficios de una red privada, sin la necesidad de invertir en la infraestructura que ésta siempre requiere. Básicamente está constituida por una conexión con la apariencia y ventajas de un enlace dedicado (Punto a punto o Frame a Relay) pero que se desarrolla a través de una red "no privada" o compartida.

Utilizando una técnica llamada "tunneling", los paquetes de información viajan a través de una red pública en una especie de "túnel privado" que simula una conexión punto a punto y que aísla dicho tráfico del resto de la red.

En principio, las redes privadas virtuales no aportan ningún concepto nuevo a lo ya existente en el ámbito de las redes de comunicación. En el caso de un usuario de acceso remoto, éste se dedica a enviar y recibir cadenas de paquetes PPP (Point-to-Point Protocol) a un servidor de acceso remoto; en el caso de un enlace entre dos redes locales, son los ruteadores de cada una de estas redes las que se intercambian los mismos paquetes PPP. La novedad de las redes virtuales privadas es que en estas conexiones los intercambios de paquetes PPP se realizan a través de una red pública de datos (por ejemplo, Internet) y no a través de enlaces directos o líneas dedicadas.

Lo que se trata es de encapsular protocolos de red ya existentes (IPX, IP Apple Talk, etc.) en paquetes PPP y éstos a su vez son encapsulados en protocolos de Tunneling, ya sean ***PPTP***, proporcionado por Microsoft en Windows 95 y Windows NT 4.0, o bien ***L2TP***. De esta manera, es posible compartir redes locales remotas a través de RDSI o ***RTB*** a costo de llamada local y sin necesidades de contratar costosas líneas dedicadas.

Las VPN corren principalmente en el ambiente IPv4, pero es importante que tengan la posibilidad de actualizarse a IPv6 para mantener operables sus soluciones VPN.

Algunos de los beneficios que proporcionan las Redes Virtuales Privadas son:

- No se requiere inversión en infraestructura, sólo se debe suscribir las líneas al proveedor seleccionado.
- Son plenamente compatibles con los esquemas de redes privadas ya implementadas.
- Ahorro en costos de llamadas de larga distancia.
- Reemplaza las costosas baterías de modems por una simple conexión a Internet.
- Permite a los usuarios remotos acceder a los recursos de la empresa a través de una simple conexión a Internet.
- Reduce los considerables costos de mantenimiento y soporte de los usuarios.
- Elimina los costos de comunicaciones que suponen los enlaces directos entre las redes de la empresa.

Los servicios proporcionados por las redes virtuales permitirán a los corporativos manejar todas sus telecomunicaciones a través de los ***carries***, sin que las empresas tengan que intervenir en infraestructura propia.

2.11. Nuevas tendencias.

Internet desde su creación en los años 60 ha tenido como aplicación principal el intercambio de información académica. En los últimos tiempos ha experimentado cambios y un crecimiento tanto de usuarios como en aplicaciones y servicios disponibles.

El comercio electrónico, la TV interactiva, la videoconferencia, los juegos, la telefonía, las comunicaciones en tiempo real, etc., son algunas de las nuevas aplicaciones y servicios que se ofrecen. Esto ha implicado el desarrollo de tecnologías, entre ellas una nueva espina dorsal llamada Internet 2, gestada por más de 150 Universidades de Estados Unidos, junto con la industria y el gobierno. Una de las principales características es el gran ancho de banda que ofrece, el canal de comunicación puede alcanzar velocidades de hasta 655 Mbps (OC-12).

Para el desarrollo de Internet 2 se creó una infraestructura de red conocida como "Abilene". Este proyecto sigue siendo dirigido por la Corporación Universitaria para el Desarrollo Avanzado de Internet (UCAID, University Corporation for Advanced Internet Development) cuyo objetivo presupone la creación de numerosas aplicaciones, cuyas funciones principales serán la seguridad, clasificación de información, transmisiones de audio y vídeo digital, almacenamiento distribuido y control de calidad automatizado de los servicios disponibles.

A esta red pueden conectarse otras universidades y negocios de todo el mundo mediante los llamados vBNS (Very High Backbone Network Service; Servicio de red de espina dorsal muy fuerte). También, se han diseñado una serie de nuevos protocolos que permiten dar soluciones a la gran demanda de servicios, tal como IPv6 (Internet Proctol version 6; Protocolo Internet versión 6). Tiene como características principales un espacio de direcciones ampliadas (de 32 bits a 128 bits), junto con unas facilidades de encaminamiento mejoradas. El espacio de direcciones no sólo aumenta, si no que su estructura se hace más adecuada para un encaminamiento óptimo mediante una jerarquía de prefijos que implican distintos niveles de asignación.

Las principales características que diferencian a Internet 2 del que se utiliza comercialmente hoy en día, son las siguientes:

- *Mayor ancho de banda.* Concede tener un canal de comunicación entre ellas de hasta 655 Mbps (OC-12).
- *Calidad de servicio* (Quality of service). Dar prioridad en la transmisión de datos de un servicio determinado.
- *Transmisión multipunto* (Multicast). Esta tecnología permite que se envíe una sola vez cada paquete con la información necesaria para que les llegue a todos los usuarios que deben recibirlo.
- *Retardo reducido* (Low Latency). Con la combinación de un gran ancho de banda, la prioridad de los servicios y técnicas avanzadas de encaminamiento se logran retardos realmente muy pequeños (del orden de milisegundos).
- *Seguridad y privacidad.* Mediante el empleo de nuevos protocolos permite autentificar plenamente el origen de los datos y asegurar la integridad y la confidencialidad de los mismos.

A continuación se enuncian algunas aplicaciones que están funcionando sobre Internet 2:

- *Nano manipulador distribuido.* La Universidad de Carolina del Norte ha desarrollado una interfaz con técnicas de realidad virtual que permite a químicos, biólogos y físicos observar a distancia la superficie de un material a escala atómica y aún más, sentir su textura a través de una palanca que cuenta con un mecanismo de retroalimentación que va dando la sensación de la textura del material a la persona que está manipulándola mientras observa en el monitor el desplazamiento que se va realizando. Utilizando esta misma palanca y oprimiendo un botón, puede modificarse la superficie. Este sistema permite tener laboratorios de microscopia virtual en los cuales pueden colaborar equipos de científicos geográficamente distantes. Esta aplicación hace uso precisamente del gran ancho de banda de Internet 2, de la prioridad de los servicios y del retardo reducido que esta red brinda.
- *Video esteroescópico a través de la red de enseñanza e investigación.* La Universidad de Carolina del Norte ha desarrollado un sistema que permite ver a distancia y en tiempo real, imágenes en movimiento en tercera dimensión.

  En cooperación con el Centro de Microscopía Electrónica del Estado de Carolina del Norte, se han realizado pruebas que demuestran la utilidad de Internet 2 al permitir obtener video estereoscópico de gran calidad y bajo costo en salones de clase y áreas de investigación remotas. En esta aplicación se hace uso principalmente de las características de gran ancho de banda y calidad de servicio, con la particularidad de que se está experimentando con modelos de control de calidad de servicios.
- *Proceso distribuido para el análisis espacial de información geográfica.* El Instituto Tecnológico de Massachussetts está trabajando en la utilización de Internet 2 para el análisis de información sobre geografía, infraestructura, demografía y uso del suelo de áreas metropolitanas mediante tecnologías de yuxtaposición espacial y visualización de datos a partir de diversas fuentes distribuidas. Estos análisis son fundamentales para la planeación urbana y la protección del medio ambiente.

  Esta aplicación requiere tanto de gran ancho de banda como de retardo reducido e incluso un alto grado de seguridad, dado que la información que se maneja puede ser empleada con malos propósitos, si es interceptada por personal no autorizado.
- *Simulación distribuida de sistemas de procesamiento de señales.* La Universidad de Cincinnati ha desarrollado un simulador de circuitos analógicos de procesamiento de señales, el cual opera sobre una red de estaciones de trabajo heterogénea modelando el comportamiento de hasta un millón de componentes. Las estaciones de trabajo pueden estar geográficamente distribuidas de manera tal que puede utilizarse una gran cantidad de ellas para simular circuitos altamente complejos. Esta aplicación es ideal para aprovechar las características de Internet 2, ya que la velocidad de la simulación es inversamente proporcional al retardo de los mensajes entre los equipos que la ejecutan.

Internet 2 es una red informática privada dedicada a los ámbitos académicos. Sin embargo, si continúa el desarrollo en la dirección acostumbrada, no ha de pasar mucho tiempo antes de que tal tecnología deje de ser un privilegio universitario.

Capítulo 3

Estándares y Protocolos de las Redes de Computadoras

En la actualidad se están llevando a cabo esfuerzos considerables a nivel mundial con el fin de publicar normas y recomendaciones que sean independientes del fabricante. Siguiendo esta tendencia, muchas organizaciones están adoptando interfaces y protocolos comunes. Nuestro objetivo en este capítulo es comprender cómo funcionan los protocolos, normas e interfaces que actualmente se utilizan en las redes de computadoras.

3.1. Protocolos.

Muchas redes están organizadas como una serie de capas o niveles, cada una construida sobre la inferior. El número de capas, el nombre, el contenido y la función de cada una pueden diferir de red a red. El propósito de cada capa es ofrecer ciertos servicios a las capas superiores de modo que no tengan que ocuparse del detalle de la implementación de los servicios.

Para llevar a cabo la comunicación de la capa *n* de una máquina con la capa *n* de otra se establece una convención, la cual está determinada por un conjunto de reglas que se conocen colectivamente como ***protocolo*** de la capa *n*. Básicamente, un protocolo es un acuerdo entre las partes que se comunican sobre cómo va a proceder la comunicación.

En la figura 3.1 se muestra un esquema de red de cinco capas. Cada capa pasa datos e información de control a la capa que está inmediatamente debajo de ella, hasta llegar a la capa más baja. Podemos observar que debajo de la capa 1 está el medio físico, en el cual, se realiza la comunicación real. Entre cada par de capas adyacentes hay una ***interfaz***. La interfaz define cuáles operaciones y servicios primitivos ofrece la capa inferior y superior. Las entidades que comprenden las capas correspondientes en las diferentes máquinas se denominan ***pares***.

El conjunto de capas y protocolos recibe el nombre de ***arquitectura de red***. Y la lista de protocolos empleados por cierto sistema, con un protocolo por capa, se llama ***pila de protocolos***.

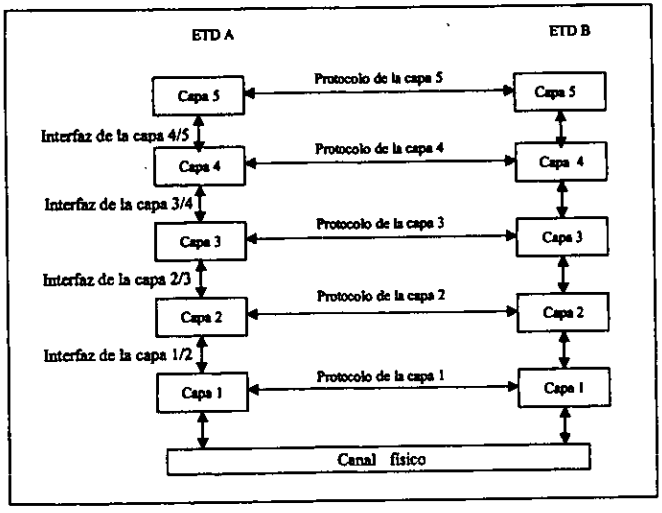


Figura 3.1. Capas, protocolos e interfaces.

Con frecuencia los conceptos de servicio y protocolo son confundidos, por ello se describen a continuación algunas características esenciales que los diferencian. El servicio es un conjunto de operaciones primitivas que definen cuáles son las tareas que la capa está preparada para ejecutar. Un protocolo es un conjunto de reglas que gobiernan el formato y el significado de los marcos, paquetes o mensajes que se intercambian entre las entidades pares dentro de una capa. Un ejemplo de ello es la siguiente analogía. Veamos un servicio como un tipo de datos abstracto (o como un objeto en un lenguaje orientado a objetos), el cual, define las operaciones que se pueden ejecutar con un objeto pero no especifica cómo se implementan éstas. Un protocolo se refiere a la implementación del servicio y como tal no es visible al usuario mismo.
A continuación se verá el modelo OSI (Open Systems Interconnetion, Interconexión de sistemas abiertos), con el propósito de explicar cómo se involucran los conceptos anteriormente vistos en un esquema de red.

3.2 Modelo de referencia OSI

La organización ***ISO*** (International Standarization Organization, Organización de Estandarización Internacional) elaboró junto con la ***ITU-T*** (en ese entonces conocido como CCITT), al final de la década de los años setentas, un conjunto de normas con el fin de conectar distintos ETDs entre sí pero con independencia del fabricante. Este conjunto de normas es conocido como el modelo de referencia OSI y los objetivos que persigue son:

- Proporcionar una serie de normas para la comunicación entre sistemas.
- Eliminar todos los impedimentos técnicos que pudieran existir para la comunicación entre sistemas.
- Abstraer el funcionamiento interno de los sistemas individuales.
- Definir los puntos de interconexión para el intercambio de información entre los sistemas.
- Limitar el número de opciones, para incrementar las posibilidades de comunicación sin necesidad de numerosas conversaciones y traducciones entre diferente productos.

La estructura de este modelo se propuso en 7 niveles o capas, como se muestra en la figura 3.3.

Capa física

La capa física tiene que ver con la transmisión de bits por un canal de comunicaciones. Las funciones incluidas dentro de esta capa son las de activar, mantener y desactivar un circuito físico entre un ETD y un ECD.
Las interfaces del nivel o capa física se utilizan para conectar dispositivos de usuario al circuito de comunicaciones y para llevar a cabo esta función se describen cuatro atributos de la interfaz. Los atributos eléctricos son los que determinan los niveles de tensión (o corriente) y la temporización de los cambios eléctricos que representan los unos

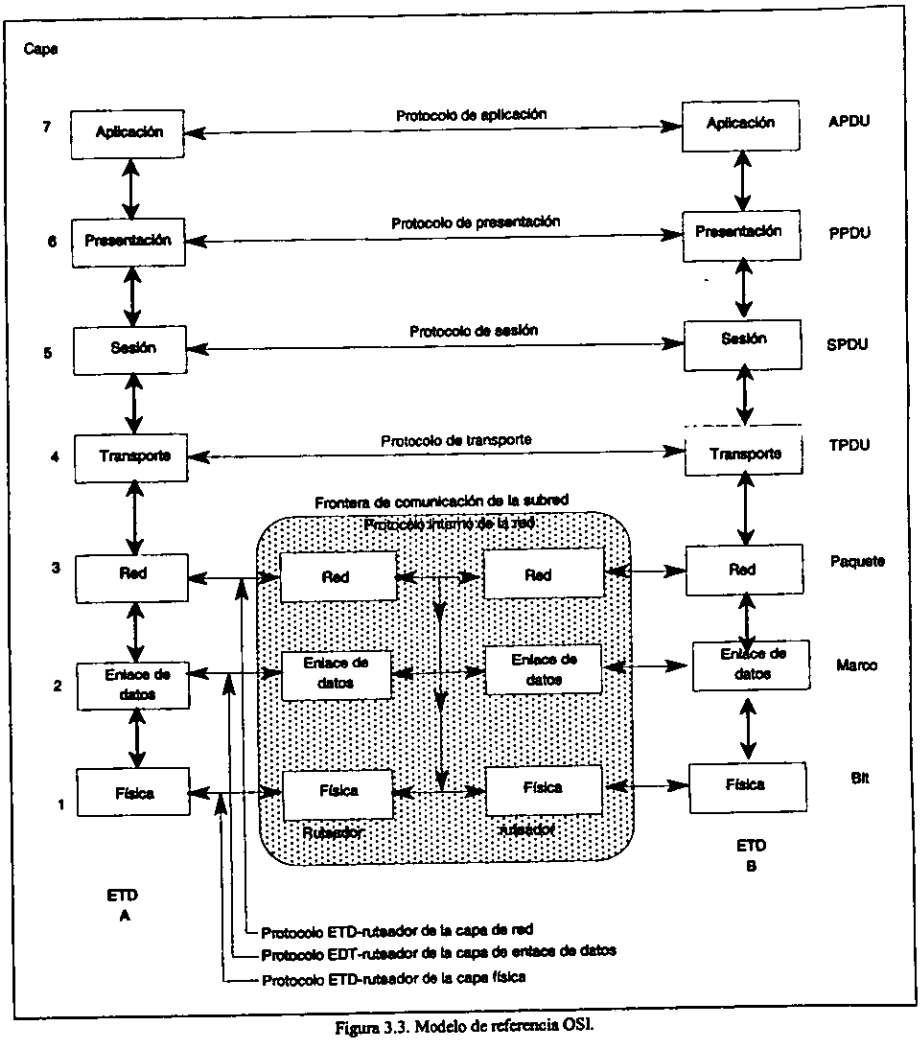


Figura 3.3. Modelo de referencia OSI.

y ceros. Los atributos funcionales incluyen el establecimiento, mantenimiento y liberación del enlace físico.

Los atributos mecánicos describen los conectores y los hilos de la interfaz. Por lo general, todas las líneas de datos, de señalización y de control están incluidas en un cable, y se conectan a enchufes terminadores situados en ambos extremos del cable. Los atributos del procedimiento describen lo que deben hacer los conectores y la secuencia de eventos necesaria para llevar a cabo la transferencia efectiva de datos a través de la interfaz.

Algunos ejemplos de interfaces que podemos encontrar en el nivel físico son ***RS-232-C*** y ***V.24.*** Ambas especificaciones describen cuatro funciones que son: Definición de las señales de control que atraviesan la interfaz, movimiento de datos, transmisión de señales de tiempo necesarias para sincronizar flujo de datos y conformación de las características eléctricas concretas de la interfaz. Sin embargo, la diferencia radica en el número de canales, ya que V.24 contiene más que RS-232-C, por lo que éste se puede considerar un conjunto del primero.

Capa de enlace

La capa de enlace es la responsable de la transferencia de datos por el canal. Entre las funciones que se llevan a cabo en esta capa son la sincronización necesaria para delimitar el flujo de bits del nivel físico; garantizar la identidad de los bits, encargándose de que los datos lleguen sin errores al ETD receptor; controlar el flujo de datos para impedir que el ETD se desborde en cualquier momento; la detección de errores en la transmisión y la recuperación de los datos perdidos, duplicados o erróneos.

En el ETD emisor se dividen los datos de entrada en marcos de datos, se transmiten en forma secuencial y se procesan los marcos de acuse de recibo que devuelve el receptor. La tarea de la capa de enlace consiste en crear y reconocer los límites de los marcos a partir de la corriente de bits proveniente de la capa física. Esto se logra añadiendo patrones especiales de bits al principio y al final del marco.

Si por alguna circunstancia el marco se destruye durante su transmisión (ráfaga de ruido en la línea), el ETD emisor no recibirá el marco de acuse de recibido y por lo tanto será una señal para indicarle al ETD emisor que genere un duplicado del marco y que lo envié de nuevo. Corresponde a esta capa resolver el problema provocado por los marcos dañados, perdidos y duplicados.

Otra consideración que surge en la capa de enlace de datos es cómo evitar que un transmisor veloz sature de datos a un receptor lento. Se debe emplear algún mecanismo de regulación de tráfico para que el transmisor sepa cuanto espacio de almacenamiento temporal (buffer) tiene el receptor en ese momento. Con frecuencia esta regulación de flujo y el manejo de errores están integrados.

- Las ***redes de difusión*** tienen una consideración adicional en la capa de enlace de datos y es cómo controlar el acceso al canal compartido. La subcapa de acceso al medio, perteneciente a la capa de enlace de datos, se encarga de este problema. El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, Institute of Electrical and Electronics Engineers) ha dividido el nivel de datos en dos subniveles llamados subnivel de control de acceso al medio (MAC, Media Access Control) y subnivel de control de enlace lógico (LLC, Logical Link Control). El subnivel MAC, el más bajo, define el método de acceso al medio, que puede ser tanto por acceso múltiple por detección de portadora y colisiones (CSMA/CD Carrier Sense Multiple Access/Collision Detection), como de anillo con testigo (Token). El subnivel LLC proporciona una forma de pasar la información entre estos diferentes tipos de red, reempaqueta los datos con nuevas cabeceras, de esta manera ofrece la funcionalidad del enlace que su propio nombre implica.

Algunos protocolos que ocupan el nivel de enlace de datos:

- Control de enlace de datos de alto nivel (***HDLC***, Hight-level Data Link Control) y protocolos síncronos orientados a bit afines
- Controladores de LAN, y métodos de acceso como Ethernet y Token Ring.
- Redes de área extensa de paquetes rápidos como Frame Realy y ATM.
- Especificación de la interfaz del controlador de red (***NDIS***, Network Driver Interface Specification) de Microsoft
- Intefaz abierta de enlace de datos (***ODI***, Open Data-link Interface) de Novell.

Capa de red

La capa de red define la interfaz entre el ETD de usuario y la red de conmutación de paquetes (subred), además de la interfaz de un ETD con otro a través de esta red. Especifica también las operaciones de ruteo por la red y la comunicación entre distintas redes.

Una consideración clave de diseño es determinar cómo se rutean los paquetes de la fuente destino. Las rutas se pueden basar en tablas estáticas que se alambran en la red y establecerse al inicio de cada conversación, por ejemplo en una sesión de terminal; o bien, pueden ser altamente dinámicas, determinándose de nuevo con cada paquete para reflejar la carga actual de la red.

Si en la subred se encuentran presentes demasiados paquetes a la vez, se estorbarán mutuamente formando cuellos de botella. El control de tal congestión pertenece también a la capa de red.

En vista de que los operadores de la subred podrían esperar remuneración por su labor, con frecuencia hay una función de contabilidad integrada a la capa de red. Cuando menos, el software debe contar cuántos paquetes, caracteres o bits envía cada cliente para producir información de facturación.

En las redes de difusión el ruteo es simple y la capa de red con frecuencia lo es más o incluso inexistente. Entre los protocolos que podemos encontrar en este nivel son:

- Protocolo Internet (***IP***)
- Protocolo ***X.25***
- Intercambio de paquetes entre redes (***IPX***, Internetwork Packet Exchange).
- Protocolo Internet VINES (VIP) de Banyan.

Capa de transporte

La función de la capa de transporte es la de aceptar los datos de la capa de sesión, dividirlo en unidades más pequeñas, pasarlos a la capa de red y asegurar que todos ellos lleguen correctamente al otro extremo.

La capa de transporte crea una conexión de red para cada conexión de transporte que requiera la capa de sesión. Sin embargo, si la conexión de transporte requiere un volumen de transmisión alto, la capa de transporte podría crear múltiples conexiones de red, dividiendo los datos entre las conexiones para aumentar el volumen. Por otro lado, si es costoso crear o mantener una conexión de red, la capa de transporte puede multiplexar varias conexiones de transporte en la misma conexión de red para reducir el costo.
La capa de transporte determina también qué tipo de servicio proporcionará a la capa de sesión y finalmente, a los usuarios de la red. El tipo de servicio se determina al establecer la sesión:

- Servicio orientado a una conexión.
- Servicio de transporte de mensajes aislados sin garantía respecto al orden de entrega.
- Servicio orientado hacia la difusión de información a múltiples destinatarios.

La capa de transporte debe cuidar de establecer y liberar conexiones a través de la red. Esto requiere alguna clase de asignación de nombres, de modo que un proceso en una máquina pueda describir con quien quiere conversar. También debe existir un mecanismo para regular el flujo de información, a fin de que un nodo rápido no sature a uno lento. Entre lo protocolos que se encuentran en la capa de transporte se encuentran:

- Protocolo de control de transmisión (TCP, Transmission Control Protocol) de Internet
- Protocolo de datagramas de usuario (UDP, User Datagram Protocol) de Internet
- Intercambio secuencial de paquetes (SPX, Sequenced Packet Exchange) de Novell
- Protocolo de comunicación entre procesos VINES (VICP, VINES Interprocess Communication Protocol) de Banyan.
- NetBIOS/NetBEUI de Microsoft

Capa de sesión

La capa de sesión funciona como una interfaz del usuario con el nivel de transporte. Coordina el intercambio de información entre sistemas mediante técnicas de conversión o diálogos. Los diálogos no son siempre necesarios, pero algunas aplicaciones pueden necesitar una forma de saber dónde reiniciar una transmisión de datos si se perdió temporalmente una conexión, o puede necesitar un diálogo periódico que indique el final de un conjunto de datos y el comienzo de otro nuevo.
Esta capa ofrece un mecanismo organizado de intercambio de datos entre usuarios. Cada usuario puede seleccionar el tipo control y de sincronización que desea de la red, como por ejemplo:

1. Diálogo bidireccional alternado o bidireccional simultáneo
2. Puntos de sincronización para comprobaciones intermedias y recuperaciones durante la transferencia de archivos.

3. Suspención y rearranques
4. Flujo de datos normal y acelerado.

Capa de presentación

Esta capa se encarga de la sintaxis y la semántica de la información de la información que se transmite. Es capaz de crear visualización de terminales virtuales. Puede también resolver la recepción de un mensaje electrónico procedente de la capa de aplicación e indicar al otro extremo que proporcione al otro nivel de aplicación un formato de página determinado.
Los protocolo de la capa de presentación forman parte del sistema operativo y de la aplicación que el usuario ejecuta en una estación de trabajo. En este nivel se da formato a la información para visualizarla o imprimirla. Los códigos se interpretan dentro de los datos, como pueden ser las secuencias de tabulación o gráficos especiales. También se gestiona el cifrado de datos y la traducción de otros conjuntos de caracteres.

Capa de aplicación

Esta capa se encarga de atender al proceso de aplicación del usuario final: transferencia de archivos, sesiones de terminales e intercambio de mensajes. Este nivel tiene en cuenta la semántica de los datos. Los protocolos de nivel de aplicación OSI:

- Terminal virtual.
- Acceso y gestión en la transferencia de archivos (FTAM, File Transfer Access and Management).
- Procesamiento de transacciones distribuidas (DTP, distributed Transaction Processing).
- Sistema de gestión de mensajes (X.400).
- Servicios de directorio (X.500)

3.3. Arquitecturas de red.

Las arquitecturas de red definen cómo se enlaza el equipo de computadoras y otros dispositivos para formar un sistema de comunicaciones que permitan a los usuarios compartir información y recursos. Existen ***arquitecturas de red propietarias***[3] como la Arquitectura de Sistemas en Red (SNA, systems Network Arquitecture) de IBM y la Arquitectura de Red Digital (DNA, Digital Network Arquitecture) de DEC, y ***arquitecturas abiertas*** como el modelo de Interconexión de sistemas abiertos (OSI, Open Systems Interconnection) definido por la Organización Internacional de Normalización (ISO, International Organization for Standardization). Si la norma es abierta,

[3] Ver en el apéndice A los temas de SNA y DECNET.

proporciona camino para que los fabricantes diseñen software y hardware interoperables con los productos de otros fabricantes. Sin embargo, el modelo OSI ha quedado como modelo con el cual se comparan protocolos y arquitecturas de red. Existen otros modelos abiertos que se están utilizando con gran éxito, es el caso de TCP/IP y ATM.

3.3.1. Modelo de referencia TCP/IP.

El modelo de referencia TCP/IP tuvo su origen junto al surgimiento de ARPANET y continua implantándose en la Internet mundial. La ARPANET era una red de investigación patrocinada por el DoD (Departamento de Defensa de Estados Unidos), la cual tenia conectada cientos de universidades e instalaciones del gobierno usando líneas telefónicas rentadas. Más tarde se añadieron redes de satélite y radio, sin embargo empezaron ha existir problemas en los protocolos para interactuar con ellas, de modo que se necesitó una nueva arquitectura con la capacidad de conectar entre sí múltiples redes. Esta arquitectura se popularizó después como el modelo de referencia TCP/IP, por las iniciales de sus dos protocolos primarios

El DoD deseaba una arquitectura cuyas conexiones permanecieran intactas mientras las máquinas de origen y destino estuvieran funcionando, aun si alguna de las máquinas o de las líneas de transmisión en el trayecto dejara de funcionar en forma repentina. También se pensaba en su flexibilidad, pues se deseaba la transferencia de archivos hasta la transmisión de voz en tiempo real.

Como se observa en la figura 3.7, este modelo está constituido por cuatro capas de red que a continuación se describen.

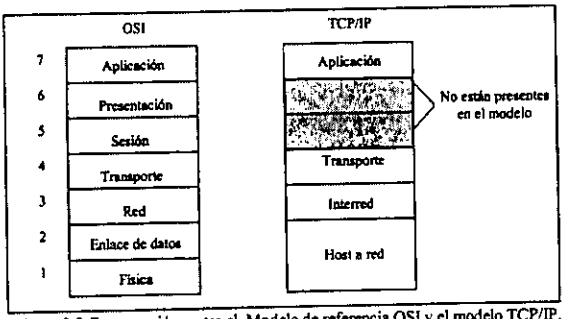


Figura 3.7.Comparación entre el Modelo de referencia OSI y el modelo TCP/IP.

Capa de interred

Esta capa, llamada capa de interred, es el eje que mantiene unida toda la arquitectura. La misión de esta capa es permitir que los nodos inyecten paquetes en cualquier red y los hagan viajar de forma independiente a su destino (que

podría estar en una red diferente). Los paquetes pueden llegar incluso en un orden diferente a aquel en que se enviaron, en cuyo caso corresponde a las capas superiores reacomodarlos, si se desea la entrega ordenada.
La capa de interred define un formato de paquete y protocolo oficial llamado IP (Internet Protocol, protocolo de interred). El trabajo de la capa de interred es entregar paquetes IP a donde se supone que deben ir. Aquí la consideración más importante es el ruteo de los paquetes, y también evitar la congestión. La capa de interred TCP/IP es muy parecida en funcionalidad a la capa de red OSI.

Capa de transporte

Esta capa se diseñó para permitir que las entidades pares en los nodos de origen y destino lleven a cabo una conversación, lo mismo que en la capa de transporte OSI.
Aquí se definieron dos protocolos de extremo a extremo. El primero, TCP (transmission control protocol, protocolo de control de la transmisión) es un protocolo confiable orientado a la conexión que permite que una corriente de bytes originada en una máquina se entregue sin errores en cualquier otra máquina de la interred. Este protocolo fragmenta la corriente entrante de bytes en mensajes discretos y pasa cada uno a la capa de interred. En el destino, el proceso TCP receptor reensambla los mensajes recibidos para formar la corriente de salida. El TCP también se encarga del control de flujo para asegurar que un emisor rápido no pueda saturar a un receptor lento con más mensajes de los que pueda manejar.
El segundo protocolo de esta capa, el UDP (user datagram protocol, protocolo de datagrama de usuario), es un protocolo sin conexión, no confiable, para aplicaciones que no necesitan la asignación de secuencia ni el control de flujo del TCP y que desean utilizar los suyos propios. Este protocolo también se usa ampliamente para consultas de petición y respuesta de una sola ocasión, del tipo cliente-servidor, y en aplicaciones en las que la entrega pronta es más importante que la entrega precisa, como las transmisiones de voz o vídeo

Capa de aplicación

El modelo TCP/IP no tiene capas de sesión ni de presentación. No se pensó que fueran necesarias, así que no se incluyeron. La experiencia con el modelo OSI ha comprobado que esta visión fue correcta: se utilizan muy poco en la mayor parte de las aplicaciones.
Encima de la capa de transporte está la capa de aplicación, que contiene todos los protocolos de alto nivel. Entre los protocolos más antiguos están el de terminal virtual (TELNET), el de transferencia de archivos (FTP) y el de correo electrónico (SMTP). El protocolo de terminal virtual permite que un usuario en una máquina ingrese en una máquina distante y trabaje ahí. El protocolo de transferencia de archivos ofrece un mecanismo para mover datos de una máquina a otra en forma eficiente. El correo electrónico fue en sus orígenes sólo una clase de transferencia de archivos, pero más adelante se desarrolló para él un protocolo especializado; con los años, se le han añadido muchos

otros protocolos, como el servicio de nombres de dominio (DNS) para relacionar los nombres de los nodos con sus direcciones de la red; NNTP, el protocolo que se usa para transferir servicios de noticias; HTTP el protocolo que se usa para recuperar páginas en la World Wide Web y muchos otros.

Capa del nodo a la red

Bajo la capa de interred está un gran vacío. El modelo de referencia TCP/IP realmente no dice mucho de lo que aquí sucede, fuera de indicar que el nodo se ha de conectar a la red haciendo uso de algún protocolo de modo que pueda enviar por ella paquetes de IP. Este protocolo no está definido y varía de un nodo a otro y de red a red.
El modelo de referencia OSI se desarrolló antes de que se inventaran los protocolos. Significa que el modelo no se orientó hacia un conjunto específico de protocolos, convirtiéndolo en un modelo muy general. El lado malo de este orden es que los diseñadores no tenían mucha experiencia con el asunto y no supieron bien cuál funcionalidad poner en cuál capa. Es por eso, que más tarde, se tuvo que insertar subcapas para eliminar algunas diferencias, pues al construir redes reales haciendo uso del modelo OSI, y de los protocolos existentes se descubrió que no cuadraban con las especificaciones de servicios requeridos.
Lo contrario sucedió con TCP/IP; primero fueron los protocolos, y el modelo fue sólo una descripción de los protocolos existentes. No hubo problema de ajustar el protocolo al modelo. El problema fue que el modelo no se ajustaba a ninguna otra pila de protocolos.
Una diferencia notoria entre los modelos de los cuales se hablaron anteriormente es en la cantidad de capas. El modelo OSI tiene siete capas y el TCP/IP cuatro. Ambos tienen capas de interred, de transporte y de aplicación.
Otra diferencia se tienen en el área de la comunicación sin conexión frente a la orientada a la conexión. El modelo OSI apoya la comunicación tanto sin conexión como la orientada a la conexión en la capa de red, pero en la capa de transporte lo hace únicamente con la comunicación orientada a la conexión. El modelo TCP/IP sólo tiene un modo en la capa de red (sin conexión) pero apoya ambos modos en la capa de transporte. Esta elección es importante sobre todo para los protocolos simples de petición y respuesta.

3.3.2. Modelo de referencia ATM.

La estructura en que se basa la ATM consiste en transmitir la información en paquetes pequeños de tamaño fijo llamados celdas. Éstas tienen una longitud de 53 bytes, de los cuales cinco son de encabezado y 48 de carga útil.
El uso de una tecnología de conmutación de celdas se debe a la flexibilidad y la facilidad en el manejo de tráfico audio, vídeo y datos. Para el manejo de velocidades muy altas (gigabits por segundo), la conmutación digital de las celdas es más fácil que el empleo de las técnicas de multiplexión, en especial si se usa fibra óptica.

Las redes ATM son orientadas a la conexión. Para hacer una llamada se debe enviar un mensaje para establecer la conexión. Después todas las celdas subsecuentes siguen la misma trayectoria al destino. La entrega de celdas no está garantizada, pero su orden sí.

Las redes ATM se organizan como las WAN tradicionales, con líneas y conmutadores (ruteadores). Las velocidades pretendidas para las redes ATM son de 155 Mbps y 622 Mbps con la posibilidad de alcanzar los gigabits. La velocidad de 155 Mbps se escogió porque es cercana a lo que se necesita para transmitir televisión de alta definición. La selección exacta es de 155.52 Mbps y se seleccionó por cuestiones de compatibilidad con el sistema de transmisión **SONET** de AT&T. La velocidad de 622 Mbps se eligió para que se pudieran mandar por ella cuatro canales de 155 Mbps.

La ISDN de banda ancha con ATM tiene su propio modelo de referencia, diferente del modelo OSI y también del modelo TCP/IP. Consiste en tres capas: la capa física, la capa ATM y la capa de aplicación de ATM, más cualquier otra cosa que los usuarios definan. Véase la figura 3.8.

Capa física

La capa física tiene que ver con el medio físico: voltajes, temporización de bits y varias consideraciones más. ATM no prescribe un conjunto de reglas en particular, pero en cambio dice que las celdas ATM se pueden enviar por sí solas por un cable o fibra o bien se pueden empacar dentro de la carga útil de otros sistemas portadores. ATM se diseñó para que fuera independiente del medio de transmisión.

La capa física se dividen en dos subcapas, una en el fondo que hace el trabajo y una subcapa de convergencia en la parte superior que proporciona la interfaz adecuada con la capa superior.

La subcapa PMD (Physical Medium Dependent, dependiente del medio físico) establece la interfaz con el cable real; y sus funciones dependen del medio físico, sea eléctrico u óptico, como son transmisión y temporización de bits.

La otra subcapa de la capa física es la subcapa TC (Transmission Convergence, convergencia de transmisión). Cuando se transmiten las celdas, la capa TC las envía como una corriente de bits a la capa PMD. En el otro extremo, la subcapa TC obtiene una corriente entrante de bits de la subcapa PMD; su trabajo es convertir esta corriente de bits en una corriente de celdas para la capa ATM. La subcapa TC se encarga de todas las consideraciones que se relacionan con determinar dónde terminan las celdas en la corriente de bits.

En el modelo OSI y en casi todas las demás redes, el trabajo de enmarcar, esto es, de convertir una corriente de bits en bruto en una secuencia de marcos o celdas, es tarea de la capa de enlace de datos.

Capa ATM

La capa ATM tiene que ver con las celdas y su transporte, define la organización de las celdas y dice lo que significan los campos del encabezado. Esta capa también tiene que ver con el establecimiento y la liberación de circuitos virtuales y aquí es donde se localiza el control de la congestión.

Capa de adaptación

Se ha definido una capa sobre la capa ATM que permita a los usuarios enviar paquetes mayores que una celda porque la mayor parte de las aplicaciones no quieren trabajar de manera directa con celdas. La interfaz ATM segmenta estos paquetes, transmite las celdas en forma individual y las reensambla en el otro extremo. Esta capa es la AAL (ATM Adaptation Layer, capa de adaptación de ATM)

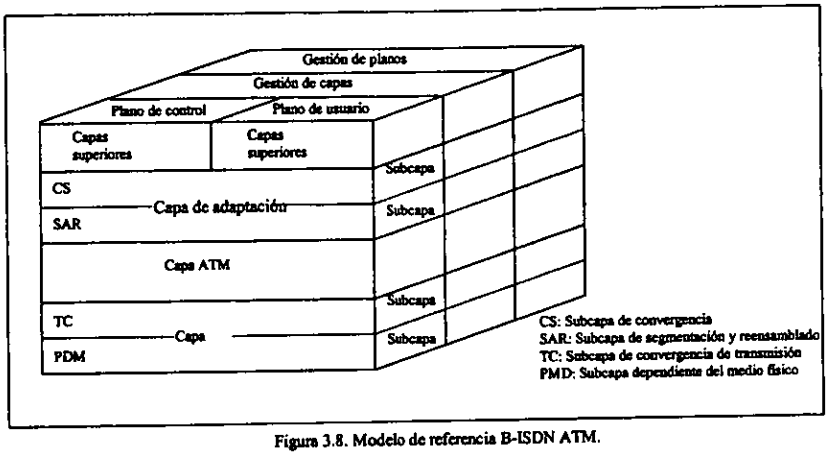


Figura 3.8. Modelo de referencia B-ISDN ATM.

La capa AAL se divide en la subcapa SAR (segmentation and reassembly, segmentación y rensamblado) y la CS (Convergence Sublayer, subcapa de convergencia). La subcapa inferior divide los paquetes en celdas en el lado de la transmisión y los vuelve a armar de nuevo en el destino. La subcapa superior hace posible tener sistemas ATM que ofrezcan diferentes clases de servicios a diferentes aplicaciones.

A diferencia de los antiguos modelos de referencia bidimensionales, el modelo ATM se define en tres dimensiones. El plano de usuario se encarga del transporte de los datos, el control de flujo, la corrección de errores y otras funciones de usuario. En contraste, el plano de control tiene que ver con la administración de la conexión. Las funciones de gestión de capas y planos se relacionan con la administración de recursos y la coordinación de intercapas. En la siguiente tabla se resumen las funciones de cada capa.

Capa ATM	Subcapa ATM	Funcionalidad
AAL	CS	Proporciona la interfaz estándar (convergencia).
	SAR	Segmentación y reensamblado.
ATM		Control de flujo. Generaciónn/extracción de encabezados de la celda. Gestión de circuitos/trayectorias virtuales. Multiplexión/desmultiplexión de celdas.
Física	TC	Desacoplamiento de la velocidad de envío de celdas. Generación y comprobación de la suma de verificación del encabezado. Generación de celdas. Empacado/desempacado de celdas de la envoltura que las encierra. Generación de marcos.
	PMD	Temporización de bits. Acceso físico a la red.

Tabla 3.1. Las capas y subcapas de ATM y sus funciones

Para obtener una mayor referencia del funcionamiento de esta arquitectura ver el apéndice A.

3.4. Norma Estándar: IEEE 802.X.

La normativa más extendida a nivel mundial en el ámbito de las LANs es la correspondiente a la familia de estándares 802.x del IEEE. Las dos topologías más utilizadas en LAN son la conexión a través de un bus de comunicación o por medio de un anillo. En este sentido, las normativas desarrolladas por el IEEE corresponden a estas dos topologías. La familia 802.x se divide en seis estándares presentados en la figura 3.9.

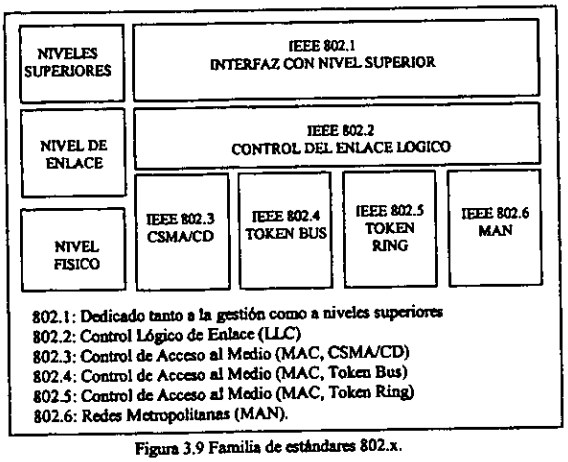


Figura 3.9 Familia de estándares 802.x.

3.4.1. La 802.2: El subnivel LLC.

El protocolo asociado al subnivel LLC presenta tres partes, la primera se refiere a la comunicación con el nivel de red; otra es la relacionada con la comunicación con el subnivel inferior, el MAC (Control de Acceso al Medio); y finalmente, la comunicación en el subnivel LLC.

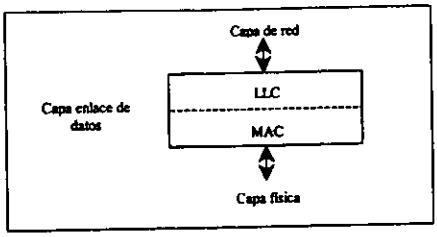


Figura 3.10. Posición del LLC.

El uso de LLC es el siguiente. En la figura 3.10, la capa de red de la máquina transmisora pasa un paquete al LLC utilizando las primitivas de acceso del LLC. La subcapa LLC entonces agrega una cabecera que contiene los números de secuencia y acuse. La estructura resultante se introduce entonces en el campo de carga útil de un marco 802.x y se transmite. En el receptor ocurre el proceso inverso.

La normativa establece que la interfaz con el subnivel MAC, se realice a través de servicios, de forma que éstos se definan con independencia del método de acceso y del canal físico empleado por el subnivel MAC. La comunicación entre el subnivel LLC y el MAC se efectúa con la ayuda de lo que se conoce como *unidades de datos de protocolo* (UDP). El formato de esta unidad de datos es el representado en la figura 3.11.

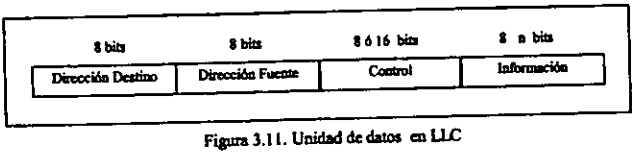


Figura 3.11. Unidad de datos en LLC

El LLC proporciona dos tipos de servicios: servicio orientado a conexión y servicio no orientado a conexión. La cabecera del LLC está basada en el antiguo protocolo ***HDLC***[4]. En el caso del servicio orientado a conexión, los marcos de datos contiene una dirección de origen, una dirección de destino, un número de secuencia, un número de acuse y datos del usuario. En el servicio no orientado a conexión, se omiten el número de secuencia y el número de acuse.

[4] Para obtener mayor información del protocolo HDLC ver apéndice A.

3.4.2. La 802.3: CSMA/CD.

El medio más probado para controlar una red local con estructura de bus es el acceso múltiple por detección de portadora y colisiones (CSMA/CD, Carrier Sense Multiple Access with Collision Detection), que puede clasificarse como un sistema sin prioridad.

Históricamente, una red con topología en bus CSMA/CD también es conocida como red tipo Ethernet. Comúnmente se usan cuatro tipos de cableado en su implementación física y éstos son los siguientes:

Nombre	Cable	Segmento máximo	Nodos/seg
10BASE5	Coaxial grueso	500 m	100
10BASE2	Coaxial delgado	200 m	30
10BASE-T	Par trenzado	100 m	1024
10BASE-F	Fibra óptica	2000 m	1024

Tabla 3.2. Tipos más comunes de LAN 802.3 de banda base

La diferencia entre los dos primeros tipos de cables coaxiales presentados en la tabla 3.2, aparte de su grosor, es la ubicación de los ***transceptores*** electrónicos (Figura 3.12). En el caso de los coaxiales gruesos, esta unidad se localiza sobre el propio cable, conocida como Unidad Transceptora y de Conexión Integrada; mientras que en los coaxiales delgados la unidad transceptora se localiza en la tarjeta de interfaz ubicada en el ETD.

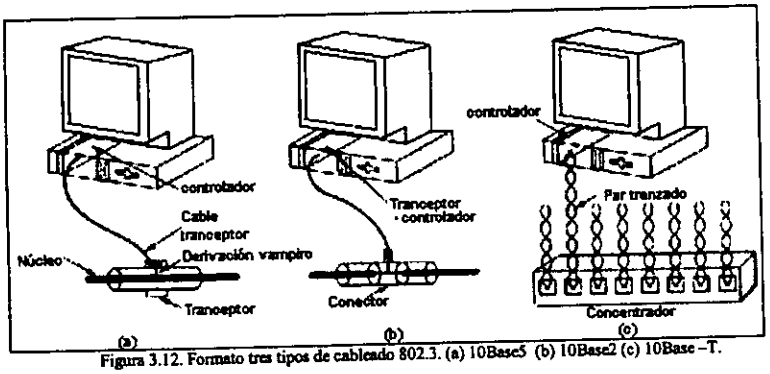


Figura 3.12. Formato tres tipos de cableado 802.3. (a) 10Base5 (b) 10Base2 (c) 10Base –T.

En el caso de cables coaxiales gruesos, la conexión física se realiza sin necesidad de cortar el cable. La Unidad Transceptora y de Conexión Integrada dispone de un mecanismo capaz de pinchar en el cable, y alcanzar por un lado el núcleo del conductor, y por otro la malla exterior.

La parte correspondiente al transceptor dispone de los dispositivos electrónicos adecuados para realizar las funciones de captación y recepción de señales; detectar colisiones; aislar el cable y el dispositivo electrónico; y proteger de sobretensiones al transceptor o al ETD.

Con 10Base5, la conexión entre el transceptor y el equipo ETD se realiza a través de pares de cables trenzados. Se dispone de cinco pares, uno de ellos permite alimentar a la unidad transceptora; dos de ellos para la transmisión de los datos (envío y recepción); y otros dos para tareas de control, uno desde el ETD y otro hacia el ETD.

En 10Base2, la conexión al cable del transceptor es sólo un conector BNC pasivo de unión T. La electrónica del transceptor está en la tarjeta controladora, y cada estación siempre tiene su propio transceptor.

Con 10BASE-T todas las operaciones se realizan sobre la tarjeta de interfaz localizada en el ETD. También se requiere de un concentrador (hub) cuyas funciones se reducen a la simple recepción y envío de las señales eléctricas.

Una cuarta opción de cableado para 802.3 es 10BASE-F que usa fibra óptica. Esta alternativa es cara debido al costo de los conectores y los terminadores, pero tiene excelente inmunidad contra el ruido y es el método más apropiado para conexiones entre edificios o entre concentradores muy separados.

Formato de la trama

La figura 3.13 muestra el formato de la trama que consiste de ocho campos. Todos los campos se transmiten mediante el código Manchester. El primer campo en ser enviado es el preámbulo, cuya longitud es de 7 bytes y la siguiente combinación de unos y ceros: 10101010. La función asociada a este campo es la de facilitar una señal que posibilite la sincronización para que los restantes campos sean recibidos correctamente. A continuación de este campo aparece el demilitador de comienzo de campo (SFD), está compuesto por un octeto cuya combinación de unos y ceros es 10101011. Los siguientes campos son de dirección, el primero corresponde al o los ETDs receptores, mientras que el segundo identifica al ETD emisor. Estos campos pueden presentar una longitud de 2 ó 6 bytes, pero en cualquier caso será fija para cada aplicación. El primer bit de la dirección destino permite indicar si se trata de una dirección individual, con lo que los datos sólo serán tomados en cuenta por un ETD; o por ETDs integrantes de un grupo. Este bit es conocido como bit I/G. Un caso especial ocurre cuando todos los bits del campo de la dirección destino se encuentran en 1, entonces decimos que la trama es de difusión, donde los datos transmitidos serán recibidos por la totalidad de los ETDs de la LAN.

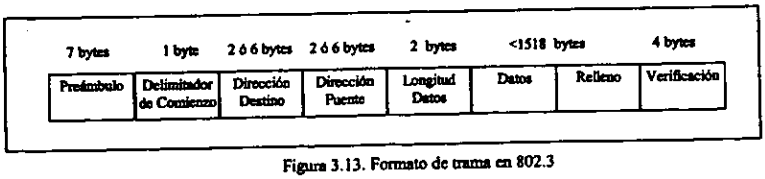


Figura 3.13. Formato de trama en 802.3

El siguiente campo recibe el nombre de indicador de la longitud y está compuesto por dos bytes, en él se codifica el número de bytes que ocupa el campo de datos. Este campo es de longitud variable, de forma que si el tamaño de toda la trama resulta ser menor de 64 bytes, se añade una secuencia de bits de relleno hasta completar el tamaño mínimo requerido. Si estos bits de relleno existen, forman parte del campo de relleno. La longitud permitida para la trama es de 1518 bytes.

El último campo presenta una longitud de cuatro bytes que permite introducir al ETD emisor el valor obtenido para una determinada función de verificación, con el fin de identificar posibles errores.

Transmisión y recepción de la trama

CSMA/CD está organizada en torno a la idea de protocolos estratificados mostrados en la figura 3.14. El nivel de usuario es atendido por los dos estratos de CSMA/CD, el de enlace y el físico. El nivel de enlace es el que proporciona la lógica que gobierna realmente la red CSMA/CD. Es independientemente del medio, y por tanto no le afecta el que la red sea de banda ancha o estrecha.

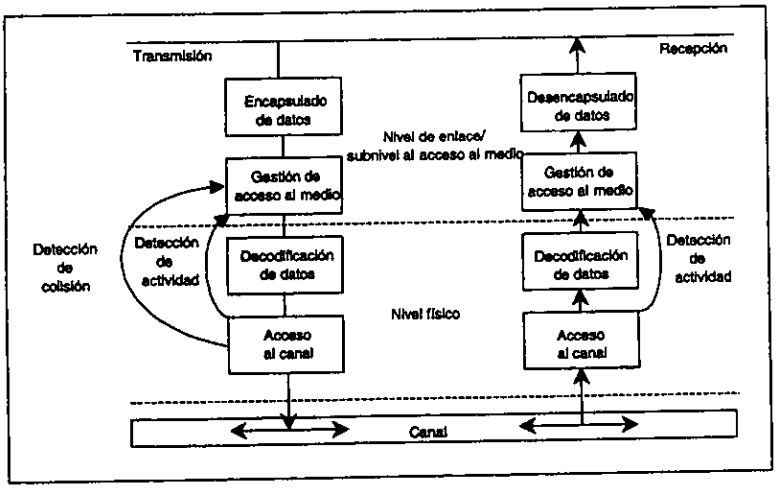


Figura 3.14. Formatos y niveles IEEE 802.3

El nivel de enlace incluye una entidad que se ocupa de encapsular y desencapsular los datos, y otra encargada de gestionar el acceso al medio, tanto para transmitir como para recibir.

Encapsulado/desencapsulado:

- Establece la trama CSMA/CD.
- Proporciona las direcciones de la fuente y del destino.

- Calcula en el nodo emisor, un campo para detección de errores y emplea ese mismo campo en el nodo receptor para indicar si ha aparecido algún error.

Gestión del acceso al medio:

Esta tarea realiza las siguientes actividades:

- Transmite la trama, y también la extrae al nivel físico.
- Almacena la trama en un buffer o memoria intermedia
- Intenta evitar colisiones (en el lado emisor).
- Gestiona las colisiones (en el lado emisor).

El nivel físico sí depende del medio. Este último se encarga de introducir las señales eléctricas en el canal; de proporcionarles el sincronismo adecuado; de codificar y decodificar los datos. Al igual que el nivel de enlace, el nivel físico está formado por dos entidades principales: la entidad de codificación/decodificación de datos, y la entidad de acceso al canal en recepción y en transmisión. Éstas son sus principales funciones.

Codificación/decodificación de datos:

- Genera las señales necesarias para sincronizar las estaciones del canal (esta señal de sincronismo se conoce como preámbulo).
- Codifica la corriente de datos binarios con un código con autosincronización en el nodo emisor, y vuelve a convertir el código Manchester en datos binarios en el receptor.

Acceso al canal:

- Introduce la señal física en el canal en el lado emisor, y toma esa señal del canal en la parte receptora de la interfaz.
- Detecta la presencia de una señal, tanto en el lado emisor como en el receptor (lo que indica que el canal está ocupado).
- Detecta las colisiones en el canal, en el lado emisor (que indican que dos señales se han interferido mutuamente).

En una red CSMA/CD, cada estación incluye una parte emisora y una parte receptora para manejar el tráfico de datos entrantes y salientes. El lado emisor se invoca cuando el usuario desea enviar datos a otro ETD de la red, y el receptor se invoca cuando el cable transporta señales dirigidas a las estaciones de la red.

La entidad de encapsulado de trama recibe los datos del usuario y construye una trama MAC, le añade también un campo de comprobación de secuencia, y la envía a la entidad de gestión del acceso al medio que la almacena en memoria intermedia hasta que el canal esté libre. El canal se considera libre cuando la entidad de acceso al medio en

emisión, situada en el nivel físico, advierte la deshabilitación de la señal piloto de detección de actividad del canal. Después de un pequeño retardo, la entidad de gestión de acceso al medio entrega la trama al nivel físico.

En el nivel físico del nodo emisor, la entidad de codificación de datos transmite la señal de sincronización (preámbulo). Además, codifica los datos binarios mediante un código Manchester con autosincronización. Posteriormente, la señal se entrega a la entidad de acceso al medio en transmisión que se encarga de introducirla en el canal.

La trama CSMA/CD llega a todas las estaciones conectadas al canal. La señal se propaga desde el nodo originario en ambas direcciones hacia los demás nodos. Una estación receptora detecta el preámbulo, se sincroniza con esa señal y activa la señal de detección de actividad del canal. A continuación, la entidad de acceso al medio en recepción entrega la señal al descodificador de datos, el cual convierte los datos de formato Manchester al formato de cadena binaria convencional, y entrega la trama al gestor de acceso al medio.

El gestor de acceso al medio en recepción guarda la trama en un buffer hasta que la entidad de acceso al canal en recepción indique que se ha desactivado la señal de detección de actividad del canal, lo que significará que han llegado todos los bits. A continuación, la entidad de gestión del acceso al medio puede entregar los datos a un nivel superior para su desencapsulado. Durante el desencapsulado tiene lugar una comprobación de errores sobre los datos, para determinar si se ha producido alguno durante el proceso de transmisión. Si no es así, se comprueba el campo de dirección para averiguar si esa trama está destinada a ese nodo. En caso afirmativo, se entrega al nivel de usuario, junto con la dirección de destino, la fuente y la unidad de datos LLC.

Una de las funciones del subnivel MAC consiste en confeccionar la trama según el esquema anterior. Antes de iniciar la transmisión, el emisor verifica el medio con el fin de detectar si se está transmitiendo alguna otra trama. Si se está realizando otra transmisión se debe esperar a que concluya. Una vez detectado que no existe actividad en el bus de comunicación se inicia la transmisión. El transceptor simultáneamente verifica la señal de recepción procedente del medio físico, ya que aún no es seguro que no se produzca algún tipo de colisión. Si este no produce, se lleva a cabo toda la transmisión y el transceptor espera la llegada de una nueva trama, desde el medio físico o desde el ETD asociado. Si por el contrario, se detecta error de colisión, inmediatamente se pone en alto la señal de detección de colisión y se procede a la transmisión de una trama jam (secuencia aleatoria de bits) que permite que todos los ETDs detecten la colisión. Posteriormente, se corta la transmisión de la trama inicial y tras un intervalo de tiempo aleatorio se reinicia la transmisión. El tiempo aleatorio de espera se establece a partir de un cierto número aleatorio de intervalos de tiempo. La duración de cada intervalo se establece igual al doble del tiempo de propagación de la señal sobre el camino más largo de todos los componentes de la red, más un cierto margen de seguridad.

Por otro lado, cada ETD activo conectado al bus que detecta la presencia de una trama inhibe nuevas transmisiones desde él para evitar colisiones. A continuación procede a la recuperación de los datos, ayudándose de los bits del campo de preámbulo. Una vez recuperados los bits correspondientes al campo de dirección destino, se determina si la trama está destinada para él o no. En caso de ser para ese ETD, se revisa el último campo de la trama el cual es la suma de comprobación. Si algunos bits de datos se reciben erróneamente la suma de comprobación con certeza estará

mal. De esta forma es posible identificar si una trama corresponde a una porción de una transmisión deteriorada tras una colisión.

3.4.3. La 802.5: Token Ring.

Un anillo consiste en un conjunto de interfaces de anillo (RIU, Ring Interface Unit), conectadas por líneas punto a punto (Figura 3.16). Cada RIU es responsable de monitorear todos los datos que pasen por ella, además de regenerar la transmisión y entregarla a la siguiente estación. Si la dirección que aparece en la cabecera de la transmisión indica que los datos están destinados a su estación, la unidad de interfaz copiará los datos y los entregará al ETD o ETDs conectados a ella.

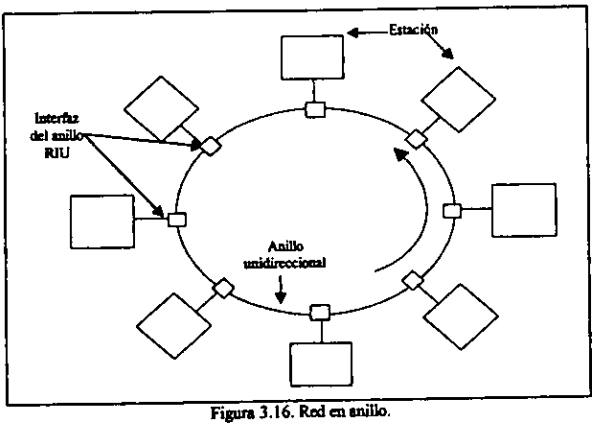


Figura 3.16. Red en anillo.

Para poder realizar la transmisión de datos se debe de disponer de una trama especial conocida como testigo (token); de tal manera que si un ETD desea transmitir debe poseer este testigo. Una vez capturado el testigo, la estación transmisora insertará datos detrás del testigo y enviará esta corriente de datos por el anillo. A medida que vayan monitoreando los datos, cada una de las RIU regenerará la señal, examinará la dirección situada en la cabecera de los datos y los transferirá a la siguiente estación. En algún momento, los datos volverán a llegar a la estación que los transmitió. En ese momento, esta estación deberá transformar el testigo ocupado en uno libre. Si el testigo vuelve a recorrer todo el anillo sin que nadie lo aproveche, la estación podrá capturarlo de nuevo y seguir transmitiendo datos. La liberación del testigo se puede producir en dos instantes de tiempo. Si la velocidad de transmisión es baja (4 Mbps) el testigo se libera tras haberse recibido la confirmación de la transmisión. Pero, si la velocidad es alta (16 Mbps), el testigo es liberado tras haberse enviado el último campo de la trama.

Un problema con las redes de anillo ocurre cuando se rompe el cable en alguna parte, con lo que el anillo se inhabilita. Esto se resuelve mediante el uso de un centro de alambrado. La red sigue siendo lógicamente un anillo,

físicamente cada estación está conectada al centro de alambrado que contiene dos pares trenzados, uno de datos a la estación y otro de datos de la estación.

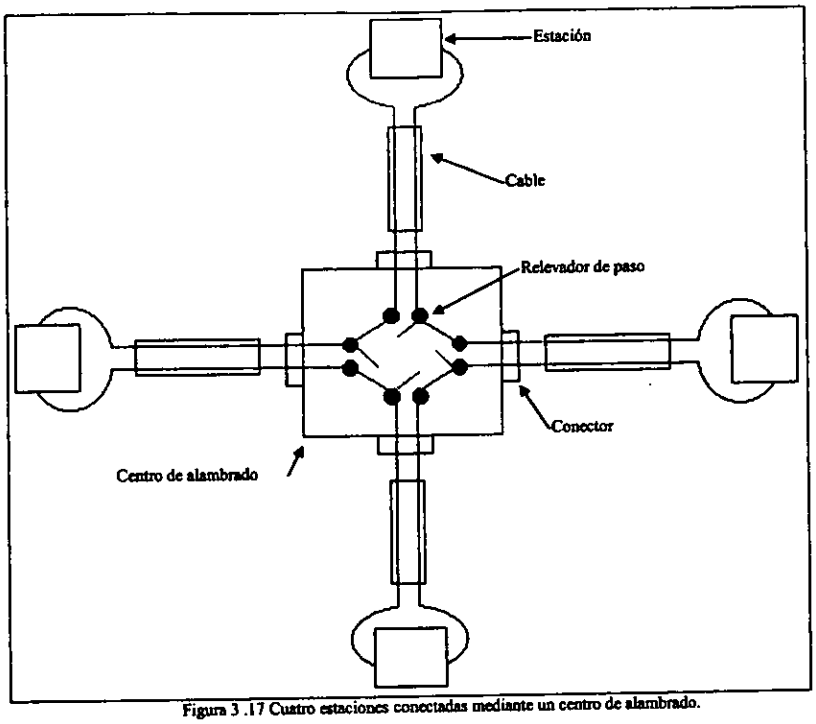


Figura 3 .17 Cuatro estaciones conectadas mediante un centro de alambrado.

Como se observa en la figura 3.17, dentro del centro de alambrado hay relevadores de paso (bypass relays) que se energizan mediante corriente de las estaciones. Si se rompe el anillo o se inactiva una estación, la pérdida de la corriente de operación desactivará el relevador, poniendo en puente la estación.

El medio físico normalmente empleado es cable de par trenzado, con el que se realiza el enlace entre dos ETDs vecinos. La conexión de cada ETD se puede efectuar de forma directa, o bien a través de un concentrador. La codificación empleada es la ***Manchester diferencial***.

Si se empleará fibra óptica como medio físico la especificación cambia ya que bajo estas condiciones existe otro tipo de red local conocida como Interfaz de Datos Distribuida por Fibra (FDDI, Fiber Distributed Interface). La descripción de esta tecnología se puede encontrar en el apéndice A.

Formato de la trama

En esta normativa aparecen dos formatos, uno para la trama asociada al testigo, y otro para la trama de transmisión de datos presentadas en la figura 3.18. La primera habilita el permiso de transmisión al ETD que en ese momento lo tiene en posesión, mientras que la segunda permite enviar datos desde un ETD a otros.

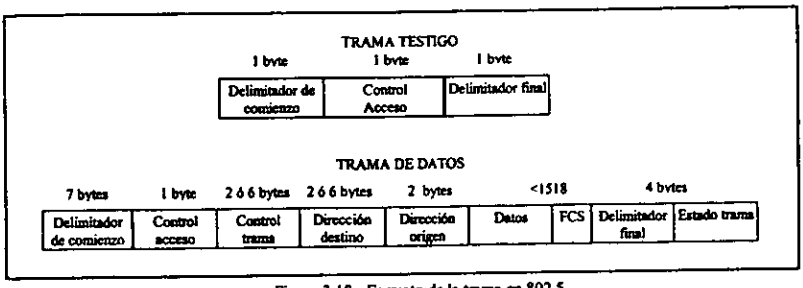


Figura 3.18. Formato de la trama en 802.5.

En los campos delimitadores de comienzo y final están formados ambos por un byte, en el que aparecen secuencias especiales de bits, con el fin de que al ser identificadas por la estación receptora, ésta pueda determinar los límites de la trama sin necesidad de establecer un tamaño fijo de trama. Esto se consigue haciendo que los bits J y K de estos campos no se codifiquen de igual forma que el resto (se utilizan patrones Manchester diferenciales no válido "HH y LL" para distinguirlos de los bytes de datos). Además, en el delimitador final los dos últimos bits son conocidos como I y E respectivamente. Si ambos adquieren el valor 0, quiere decir que la trama es de tipo testigo. Para una trama normal, es posible distinguir si se trata de la última o única de todo un mensaje (I=O), o no (I=l). El bit E se usa para la indicación de detección de errores. Inicialmente este bit es puesto al valor 0 por la estación emisora, pero si alguna de las estaciones intermedias o la destino detectan algún error lo modifican a 1. De esta forma la estación emisora conoce si se ha detectado algún error en la trama enviada. Observe la siguiente figura.

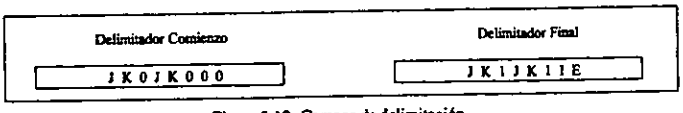


Figura 3.19. Campos de delimitación.

En el campo de control de acceso mostrado en la figura 3.20, cada bit o conjunto de ellos permite indicar prioridades, tipo de trama, reservas o monitorizaciones. Los tres primeros son conocidos como bits P o de prioridad, y permiten indicar la prioridad asociada al testigo en ese momento, indicando que estaciones tienen derecho a transmitir. El bit T se emplea para identificar a la trama, si esta a 1 indica que se trata de un testigo, si su valor es 0

entonces la trama es de datos. Cuando la trama es de testigo puede ocurrir que la información se deteriore, pudiendo dar lugar a que un testigo circulase indefinidamente por el anillo. Con el fin de evitar tales situaciones se le asigna a una de las estaciones el papel de monitor activo. Inicialmente, el testigo debe estar con el bit M a 0, de forma que cuando se haya en posesión de la estación que actúa como monitora, ésta modifica el valor de este bit a 1. Si el testigo nuevamente retorna a ella y posee el bit M a l, inicia un proceso de purga, mediante el que se le indica al resto de estaciones que el proceso va a ser reiniciado. Con lo que respecta a los últimos 3 bits R o de reserva ofrecen la posibilidad a las estaciones de más alta prioridad al valor existente en ese momento, solicitar el uso posterior del testigo.

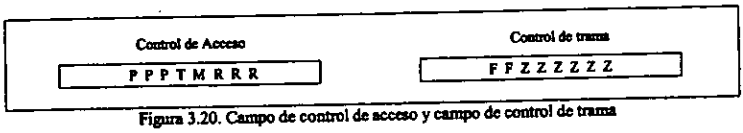


Figura 3.20. Campo de control de acceso y campo de control de trama

El campo de control de trama únicamente aparece en las tramas de datos. En él se indica el tipo de trama y algunas funciones de control. Los dos primeros bits, representados como F en la figura 3.20, indican el tipo de trama, MAC o de información. El resto, los Z, se destinan a funciones de control. Una trama de tipo MAC es interpretada por todas las estaciones, mientras que una de información sólo se interpreta por aquellas a las que va destinada la información. Los campos de direcciones, fuente y destino, pueden presentar longitudes de dos o seis bytes manteniéndose fija la longitud empleada. El primer bit correspondiente al campo de dirección destino permite indicar si se trata de una dirección individual (0) o de grupo (1). Si todos los bits de la dirección destino están a l, la trama es de difusión, y por tanto va destinada a todas las estaciones que pertenecen al anillo. El campo de información es el lugar donde se ubican los datos a transmitir. La naturaleza de los datos puede ser bien de información para niveles superiores, o de información de control para el nivel MAC cuando la trama es de tipo MAC, según la especificación del tipo de trama en el campo de control de trama. El campo FCS realiza funciones de comprobación de errores, y el campo de estado de la trama sirve para indicar que la estación receptora ha reconocido su dirección y ha copiado sus datos en el campo de información. Por último, el campo de estado de la trama (FS), mostrado en la figura 3.21, está compuesto a su vez por dos campos. Los bits de reconocimiento de dirección (A), y los de copia de datos (C). Inicialmente ambos poseen el valor 0, dado por la estación emisora. Cuando la estación destino reconoce su dirección modifica el valor del bit A a 1, y si además realiza una copia de los datos, también altera el valor del bit
C a 1. De esta forma, la estación emisora conoce si la estación destino existe o está activa, y en caso de estar activa, si ha realizado copia de los datos.

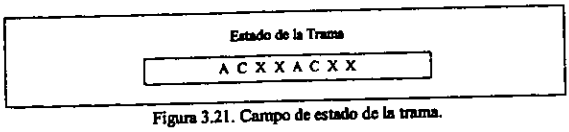


Figura 3.21. Campo de estado de la trama.

Transmisión y recepción de la trama

Una vez que la estación a transmitir ha configurado la trama con el esquema indicado anteriormente, pasa a realizar el proceso de captura del testigo que habilite la transmisión de la trama en espera. Cuando una trama testigo es recibida por esta estación con una prioridad menor o igual a la de la trama confeccionada, se puede proceder a la transmisión. Con el fin de permitir que el anillo pueda ser utilizado por estaciones con distinta prioridad, se emplea el siguiente algoritmo. Cada vez que una trama, testigo o de datos, llega a una estación que desea transmitir, verifica el valor de reserva en el campo de control de acceso. Si es igual o superior a la prioridad asociada a la trama a transmitir, la trama en circulación no sufre ninguna modificación. Si por el contrario, el valor de reserva resulta ser menor que la prioridad de la trama en espera, el valor de reserva del campo de control de acceso de la trama en circulación adquiere la prioridad asociada a la trama en espera. Si no existe ninguna otra estación posterior que modifique el valor de reserva de dicho campo, la estación que ha efectuado la reserva recibirá un testigo con su prioridad que le habilita para realizar la transmisión. En estas condiciones, dicha estación pasa a ser la emisora, con lo que efectúa las modificaciones necesarias sobre determinados bits de la trama testigo, y añade los campos que permiten configurar a la trama como de datos. A continuación, cuando la trama nuevamente retorna, puede seguir transmitiendo la misma estación más tramas, debido a que presenta mayor prioridad, y por que se encuentren dentro del intervalo de tiempo asignado a la estación para realizar transmisiones. Una vez concluido la transmisión, la estación es la encargada de restituir el valor del testigo al anillo con un valor de prioridad adecuada, que garantice, primero que las tramas de más prioridad que la existente en ese momento en el anillo sean transmitidas en primer lugar, y segundo, que todos los ETDs con la misma prioridad dispongan del mismo derecho a la hora de acceder al medio.

La condición anteriormente establecida obliga a que el testigo sea restituido al anillo con los valores de prioridad y reserva previos a la transmisión de la trama efectuada, y que obviamente presentaba mayor prioridad. Para ello la normativa 802.5 aconseja el empleo de dos grupos de valores en cada ETD. El primero está formado por los asociados a tres variables: Pm, Pr y Rr. La primera, Pm, permite almacenar el valor más alto de prioridad entre todas las tramas que están en espera a ser transmitidas en el ETD. Las variables Pr y Rr son conocidas como registros de prioridad, y almacenan los valores de la prioridad y reserva, respectivamente, que presentaba la última trama, testigo o de datos, que pasó por ese ETD. El segundo grupo de valores corresponde a dos pilas conocidas como Sr y Sx.

Una vez concluida la transmisión, el testigo ha de restituirse al anillo operando del siguiente modo. Se genera un testigo en el que la prioridad es igual a la almacenada en Pr, siempre y cuando la estación no disponga de una trama de más alta prioridad, o bien, el valor de reserva que está almacenado en Rr sea inferior a Pr. En caso contrario, se generaría un testigo con valor de prioridad mayor entre el Pm y el Rr. El valor de la reserva del testigo ha restituir, es el más grande entre Pm y Rr para el primer caso planteado anteriormente, mientras que se le asigna el valor 0 en el segundo supuesto. En el segundo caso, se puede observar que la estación va a restituir un testigo al anillo con una

prioridad superior a la que disponía cuando ella lo retomó, por lo que la prioridad primitiva es almacenada en una pila, y más concretamente en la Sr. Mientras que la nueva prioridad se almacena en la pila Sx. Estos valores son almacenados en el ETD, y es obligación suya el restituir al anillo la prioridad de Sr cuando no haya transmisión en el anillo con una prioridad mayor o igual a la de Sx. El empleo de una pila en lugar de un registro está justificada ante la posibilidad de recurrencia en el proceso. Puesto que es el ETD el encargado de almacenar y realizar estas últimas operaciones, se le conoce como estación depositaria. Asumiendo que existe la estación depositaria, cuando a ella llegue nuevamente un testigo, deberá de comprobar nuevamente las prioridades, y determinar si la prioridad asociada al testigo debe disminuir, mantenerse o aumentar. Si el valor de reserva de ese nuevo testigo es mayor que el almacenado en Sr, la prioridad asignada será Rr (igual al valor de reserva que poseía el testigo), y la prioridad que tenía asociada se introduce en la pila Sr. Por tanto, la estación continua ejerciendo de depositaria. Si por el contrario, el valor de los bits de reserva resulta ser menor o igual que el de Sr, los valores almacenados en Sx y Sr son introducidos en los bits de prioridad y reserva respectivamente. En este supuesto, las pilas Sx y Sr se encuentran vacías, y por tanto, la estación deja de ejercer como depositaria.

El proceso de recepción de una trama de datos resulta algo más simple. La estación que recibe una trama de datos debe determinar, si la información solo es nuevamente transmitida o también debe ser copiada. Si el bit F del campo de control de trama indica que es de tipo MAC, los bits C son copiados e interpretados. Si se tratase de una trama de datos, se copian los datos, y también se alteran los valores de los bits A y C del campo de estado de la trama.

Mantenimiento del anillo

Durante el proceso de inicialización del anillo, o también cuando una estación está desconectada en ese momento y desea pasar a formar parte del anillo, es necesario efectuar un proceso conocido como inicialización. Esto comienza con el envío de una trama especial conocida como DAT (Duplicate Address Test), que es de tipo MAC, y con los bits A del campo de estado de trama a 0. El propósito de esta trama consiste en determinar si la dirección con la que la estación desea incorporarse al anillo existe. Cada una del resto de estaciones activas en el anillo verifica si la dirección enviada es igual a la suya, en cuyo caso modifica el valor del bit A del campo de estado de trama al valor 1. Si una trama retorna con el bit A a 1, el ETD deberá modificar el valor de su dirección, y volver a intentar su incorporación transcurrido un tiempo. Si por el contrario, el valor es 0, deberá continuar con el proceso de inicialización. Para ello transmite una nueva trama especial conocida como SMP (Standby Monitor Present), también de tipo MAC. La finalidad es indicar a la estación sucesora la dirección de su nueva predecesora en el anillo. Una vez completada esta operación, el nuevo ETD puede proceder de forma similar al resto. Es decir, puede emitir y recibir tramas. Las funciones de monitorización corresponde a uno de los ETDs acitvos del anillo. El propósito es evitar que existan tramas que circulen permanentemente por el anillo. Para ello una de las estaciones adquiere el papel de monitor activo, con lo que debe realizar ciertas funciones de chequeo. Sin embargo, la estación que ejerce como monitor activo debe comunicar en intervalos regulares de tiempo al resto de estaciones que está activa y que continua

ejerciendo su función. Este proceso se realiza mediante la transmisión de una trama especial de tipo MAC, la AMP (Active Monitor Present). Sin embargo, puede ocurrir que el ETD que ejerce de monitor activo falle, con lo que se hace necesario emplear algún procedimiento que determine un nuevo monitor activo. Si una estación desea ser el monitor activo, inicialmente debe realizar un proceso de purga, que permita asegurar que no existe ninguna trama circulando por el anillo. Este proceso se realiza con la ayuda de una nueva trama de tipo MAC denominada PRG (Purga), y que elimina todas las demás. Si esta trama retorna a la estación que la envío, procede a informar al resto que es ella la nueva estación monitor activo. Para ello emplea una nueva trama de tipo MAC, denominada CT (Claim Token).

Una función de mantenimiento que no puede manejar el monitor es la localización de rupturas en el anillo. Cuando una estación nota que cualquiera de sus vecinas parece estar muerta, transmite un marco indicación o faro (Beacon) dando la dirección de la estación que tal vez está muerta. Cuando la señal se ha propagado, es posible que se detecte cuantas estaciones están desactivadas y eliminarlas del anillo usando los relevadores de paso del centro del alambrado.

3.4.4. La 802.4: Token Bus.

La red token bus consiste de un conjunto de estaciones organizadas lógicamente en forma de anillo, conectadas a un cable lineal. Cada estación conoce la dirección de la estación anterior y siguiente. Observe la siguiente figura.

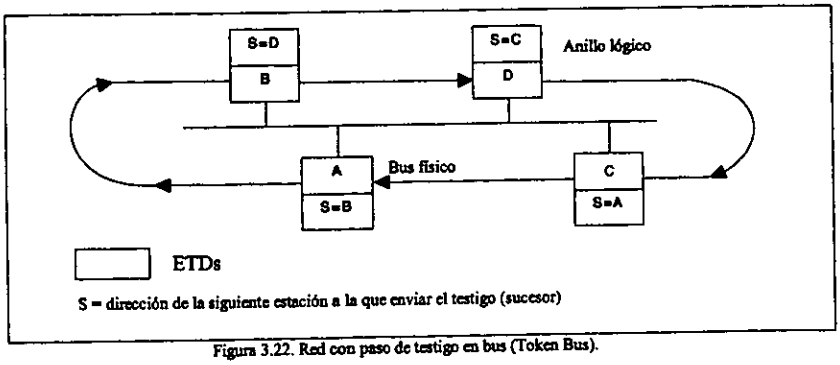


Figura 3.22. Red con paso de testigo en bus (Token Bus).

El funcionamiento de este tipo de redes también requiere del permiso de transmisión (como en una red Token Ring), mediante la posesión de un testigo. El protocolo usa una trama de control llamada testigo de acceso o derecho de acceso, que confiere a una estación el uso exclusivo del bus. La estación que posee el testigo usará el bus durante un periodo de tiempo para enviar y recibir datos y a continuación se lo entregará a otra estación designada. Mediante la

topología de bus, todas las estaciones escuchan y reciben el testigo de acceso, pero la única estación que queda autorizada para usar el canal es aquella que aparezca indicada en el testigo de acceso. Todas las demás estaciones deberán esperar su turno para recibir el testigo. Las estaciones van recibiendo el testigo cíclicamente, con lo cual se configura un anillo lógico sobre un bus físico.

El medio físico que se utiliza en estas redes es el cable coaxial. La interfaz a nivel físico realiza operaciones, tales como, codificación de los datos a transmitir, decodificación de los datos recibidos, y generación de señales de reloj. La transmisión de señales se codifica mediante el método carrierband. Consiste en transmitir la señal correspondiente a 1 binario como un ciclo simple de una señal senoidal de frecuencia igual a la velocidad de transmisión (entre 1, 5 y 10 Mbps); mientras que un 0 binario se transmite como dos ciclos de una señal senoidal de frecuencia el doble de la velocidad de transmisión. En la figura 3.23 se muestra la codificación mediante carrierband. Una de las ventajas que ofrece este método es la inmunidad al ruido. Las señales de ruido están compuestas por un número infinito de componentes de frecuencia. Como en el sistema transceptor tan sólo pasan dos frecuencias, las correspondientes a la codificación carrierband, se filtran todas las posibles señales de ruido ofreciendo de esta forma inmunidad del sistema frente al ruido.

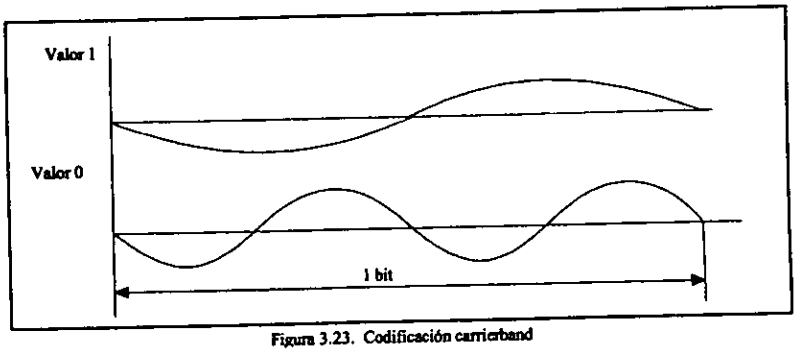


Figura 3.23. Codificación carrierband

Formato de la trama

En la figura 3.24 se presenta el formato de la trama 804.4, la cual es similar al de la especificación 802.5. La diferencia entre ellas está en el uso del preámbulo y la desaparición de los campos de control de acceso y estado de trama. Los bits J y K, existentes en los campos delimitadores de comienzo y final de la 802.5, se sustituyen en la codificación carrierband por un par esencial de símbolos que representan la ausencia de datos, representados en la figura 3.25.

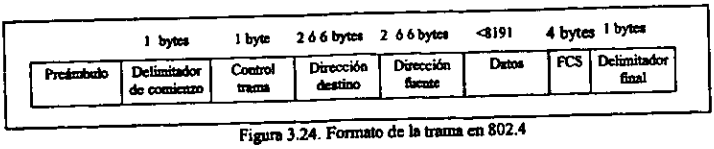


Figura 3.24. Formato de la trama en 802.4

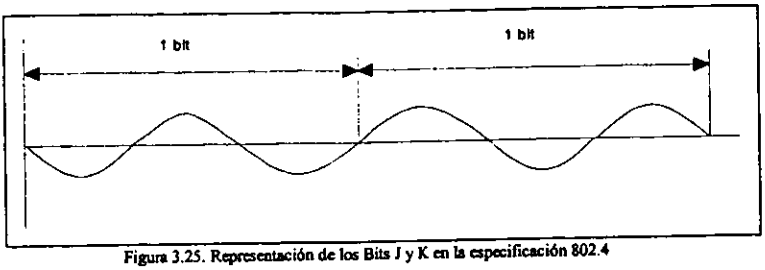
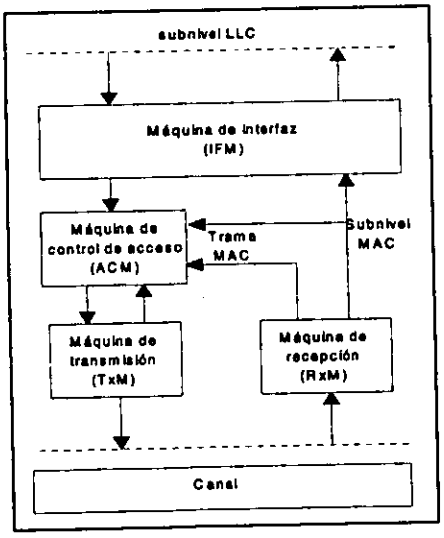


Figura 3.25. Representación de los Bits J y K en la especificación 802.4

Funcionamiento

El subnivel MAC consta de cuatro funciones principales: la máquina de interfaz (IFM), la máquina controladora de acceso (ACM), la máquina receptora de interfaz (IFM), la máquina receptora (RxM) y la máquina de tránsito (TxM). Para una mejor referencia, observe la siguiente figura.

Figura 3.26. Paso de testigo en bus (IEEE 802.4).



El corazón del sistema token bus es la máquina ACM. Determina cuándo puede colocarse una trama en el bus, y coopera con las ACM de otras estaciones para controlar el acceso al bus compartido. Asimismo, se encarga de inicializar y mantener el anillo lógico, lo cual incluye la detección de errores y la resolución de averías.

Las tramas LLC se entregan a la ACM a través de la máquina de interfaz (IFM). Este componente guarda en memoria intermedia las solicitudes LLC. La IFM manipula una serie de parámetros para optimizar la calidad del servicio desde el nivel LLC hasta el nivel MAC, y también comprueba las direcciones de las tramas LLC recibidas.

Los componentes TxM y RxM tienen misiones algo limitadas. Es responsabilidad de la TxM la transmisión de la trama al nivel físico. Acepta una trama de la ACM y construye con ella una unidad de datos del protocolo MAC (UDP), colocando al principio de la trama un preámbulo y un delimitador de comienzo (SD). Asimismo, añade al final de la trama un FCS y un delimitador de final (ED). RxM, por su parte, acepta los datos del nivel físico, e identifica que ha llegado una trama completa cuando detecta el SD y el ED. También comprueba el campo FCS para asegurar de que la transmisión está libre de errores. Por otra parte, la trama proveniente del medio pasa al componente RxM y posteriormente al IFM, del cual informa de su llegada y se la entrega al subnivel LLC.

Por otra parte, el testigo se pasa físicamente a través del bus, entonces es recibido por todos los ETDs, aunque sólo uno de ellos tienen derecho de tomarlo para transmitir, y es el establecido por el anillo lógico. Para la implementación de un anillo lógico, cada ETD debe conocer su sucesor y percheros en el anillo. Por otro lado, el medio físico es un bus, por lo que debe existir un intervalo de tiempo de transmisión para asegurar que la información ha llegado a todas las estaciones, y que define como el doble de la suma del tiempo en viajar la señal desde un extremo a otro del medio, más el tiempo de procesamiento que emplea un ETD en generar una respuesta adecuada.

La posesión de testigo de un ETD a otro, emplea los valores numéricos de las direcciones, de forma que el anillo se establece en orden descendente de ellas. Una vez enviado el testigo, el ETD pasa a escuchar las señales que se transmiten por el medio, con el fin de asegurarse que su sucesor está activo y ha recibido el testigo. Si el ETD detecta que se están transmitiendo tramas válidas, da por supuesto que el testigo ha sido recibido por su estación sucesora. Pero, si durante el intervalo de tiempo de transmisión no se detecta ninguna trama en el medio, el ETD que emitió el testigo debe realizar alguna operación correctora. También puede ocurrir que el ETD detecte que se están transmitiendo tramas, pero éstas no resultan válidas. En esta nueva situación se establece un intervalo de espera superior a cuatro veces el intervalo de tiempo de transmisión, de forma que si no se detecta nada durante este tiempo, se asume que el testigo no se recibió correctamente, y por tanto, se transmite nuevamente. Si por el contrario, durante este intervalo de espera se detecta alguna trama válida, se asume que su sucesor posee el testigo. De igual forma, si la trama no resulta válida, se considera como una transmisión realizada por el sucesor, de forma que se asume también que el sucesor posee el testigo.

Considerando las anteriores situaciones, se realiza el paso de un testigo desde un ETD a su sucesor. Sin embargo, si el sucesor no se encuentra activo, el ETD que en ese momento posee el testigo, una vez determinado que su sucesor no responde, debe pasarlo a otra estación, que ya no será su sucesor primitivo. Para solucionar esta posible situación se establece un procedimiento conocido como establecimiento de nuevo sucesor. Para ello, el ETD emite una trama conocida como «quien me sigue» (who folows me), en la que en el campo de datos se ha introducido la dirección de su sucesor hasta en ese momento. Esta trama es recibida por la totalidad de ETDs activos, con lo que cada estación compara esta dirección con la de sus predecesores. Aquel ETD que comprueba que ambas direcciones son

coincidentes, envía una nueva trama especial conocida como establecimiento de sucesor (Set successor), en la que indica su dirección.. Sin embargo, puede ocurrir que el ETD sucesor de su sucesor también esté inactivo, con lo que no recibirá respuesta a la trama quien me sigue. La forma de operar en este caso consiste en un reenvío de la trama quien me sigue, de forma que si nuevamente no existe respuesta, se realiza el envío de una nueva trama especial conocida como solicitud de sucesor, en la que se indica en el campo de dirección destino su propia dirección, y que permite el restablecimiento del anillo lógico. Las estaciones que se encuentran operativas responden, mediante un procedimiento conocido como ventana de respuesta. Si ninguna estación llegase a responder, quiere decir que ha ocurrido un fallo (ruptura del cable, fallo en el propio ETD, etc.). En este supuesto el ETD permanece en silencio, pero a la escucha de las transmisiones que pueden producirse. El procedimiento de ventana de respuesta se sigue a intervalos de tiempo aleatorio con el fin de permitir la incorporación de nuevos ETDs al anillo. Mientras un ETD está en posesión del testigo, envía una trama de solicitud de sucesor con los dos campos de direcciones especificados. La trama es respondida por todo ETD cuya dirección esté comprendida en el intervalo definido por los campos de direcciones de la trama de solicitud de sucesor. Una vez enviada la trama de solicitud de sucesor, el ETD espera un intervalo de tiempo igual al intervalo de tiempo de transmisión. Cuando esto se realiza se dice que se abre una ventana de respuesta. Si existe un ETD que desea entrar, debe enviar una trama de establecer sucesor, de forma que será el próximo en recibir el testigo. También es posible que un ETD se inhiba de recibir el testigo, para lo que es necesario que envíe una trama de establecer sucesor a su predecesor. Con ello le ordena que en lo sucesivo el testigo no le sea pasado a él sino a su sucesor.

Funcionamiento con prioridades

Las principales aplicaciones del método de acceso en token bus se encuentran en el terreno del control y la automatización de procesos industriales. En este caso sólo se distinguen cuatro niveles de prioridades, llamadas clases de acceso o de servicio:

- Clase 6: dedicada para mensajes de tipo urgente tales como condiciones de alarma crítica o funciones de control relacionados con ella.
- Clase 4: mensajes de control normales y funciones de mantenimiento del anillo.
- Clase 2: mensajes relacionados con cargas de datos.
- Clase 0: mensajes en general de baja prioridad.

Cada estación puede visualizarse como cuatro subestaciones, cada una con un nivel de prioridad. A medida que llegan las entradas desde arriba a la subcapa MAC, los datos examinan para ver su prioridad y se encaminan a una de las cuatro subestaciones. Por tanto cada subestación mantiene su propia cola de tramas a transmitir.

Cuando el token llega a la estación a través del cable, se para internamente a la subestación de prioridad 6, que puede empezar a transmitir tramas si tiene. Hecho esto (o al expirar el temporizador), el token se pasa internamente a la subestación de prioridad 4, que entonces puede transmitir tramas hasta que expire el temporizador, en cuyo momento

el token se pasa internamente a la subestación de prioridad 2. Este proceso se repite hasta que la subestación ha enviado todas sus tramas o hasta que expira el temporizador. En ambos casos, en este punto el token se pasa a la siguiente estación del anillo.

3.5. Protocolo TCP/IP

Las siglas TCP/IP provienen de los nombres de dos protocolos: Protocolo de Control de Transporte (TCP, Transport Control Protocol) y del Protocolo Internet (IP, Internet Protocol). Su diseño tuvo como objetivo la conexión entre múltiples redes. En sí, el protocolo TCP/IP es una colección de protocolos cooperativos y complementarios. Entre los protocolos más importantes están:

- **IP.** El Protocolo Internet es un protocolo de la capa de red orientado a servicios sin conexión. La función de esta capa es inyectar paquetes en cualquier red y mantenerlos viajando hasta su destino.
- **TCP.** El Protocolo de Control de Transporte es un protocolo de la capa de transporte que mueve la información entre las aplicaciones. Proporciona comunicación dividida en mensajes, servicio mediante conexión y con corrección de errores.
- **UDP.** El Protocolo de Datagrama de Usuario es otro protocolo de la capa de transporte. A diferencia de TCP, proporciona comunicación sin conexión y sin corrección de errores. Utilizado en aplicaciones de pregunta/respuesta.

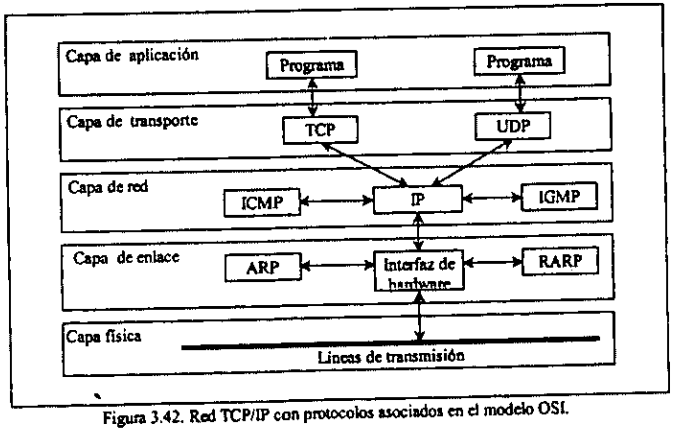


Figura 3.42. Red TCP/IP con protocolos asociados en el modelo OSI.

Una red que utiliza TCP/IP para establecer enlaces con otras redes, cubre cinco capas del modelo OSI. En el esquema de la figura 3.42, se observa que aparte de los tres protocolos mencionados, existen otros que TCP/IP requiere para la transferencia de datos a través de la red: ICMP, IGMP, ARP y RARP. Más adelante se describirán estos protocolos.

Capa física

La capa física en una red TCP incluye al medio de transmisión a través del cual se transportan los datos por la red. Este medio puede ser cable coaxial, par trenzado y fibra óptica.

Capa de enlace

La capa de enlace incluye una interfaz de hardware y dos módulos de protocolos: el Protocolo de Resolución de Direcciones (ARP, Address Resolution Protocol) y el Protocolo de Resolución de Direcciones Inverso (RARP, Reverse Address Resolution Protocol).
ARP traduce las direcciones de la capa de red a direcciones de la capa de enlace. A la inversa, RARP traduce las direcciones de la capa de enlace a direcciones de la capa de red.
El propósito de la capa de enlace es ocultar a la capa de red los detalles de la implementación física. Es por eso, que la capa de red no se preocupa si la red está utilizando tecnología Ethernet, Token ring , FDDI.

Capa de red

La capa de red incluye el Protocolo Internet (IP), el Protocolo de Control de Mensajes de Internet (ICMP, Internet Control Message Protocol) y el Protocolo de Manejo de Grupos de Internet (IGMP, Internet Group Management Protocol). ICMP e IGMP son protocolos de apoyo para el manejo de mensajes, como los de error y de transmisiones múltiples (mensajes enviados a dos o más sistemas).

Capa de transporte

Esta capa incluye el Protocolo de Control de Transporte (TCP, Transport Control Protocol) y el Protocolo de Datagrama de Usuario (UDP, User Datagram Protocol). El primero es un protocolo orientado a conexión que utiliza un flujo de bytes confiable para enviar y recibir datos. Además proporciona un circuito virtual para comunicaciones en red. El segundo es un protocolo sin conexión, no confiable, que utiliza datagramas para enviar y recibir datos.

Encapsulamiento

Conforme los datos viajan en la pila de protocolos, cada capa trabaja sobre lo que anexa la capa anterior. Observe la figura 3.43. En la capa de aplicación se encapsula los datos del usuario en un mensaje de la aplicación.

Posteriormente, en la capa de transporte, TCP formatea los datos de la aplicación en un segmento TCP, el cual incluye los datos de la aplicación junto con el encabezado de la información TCP que necesita el protocolo. Conforme la información pasa a la capa de red, el software de red formatea el segmento TCP para convertirlo en un datagrama (o paquete) IP. Finalmente, el controlador del hardware del medio físico formatea los datos del datagrama y los coloca dentro de la trama de la tecnología utilizada (por ejemplo: Ethernet, Token Ring y FDDI).

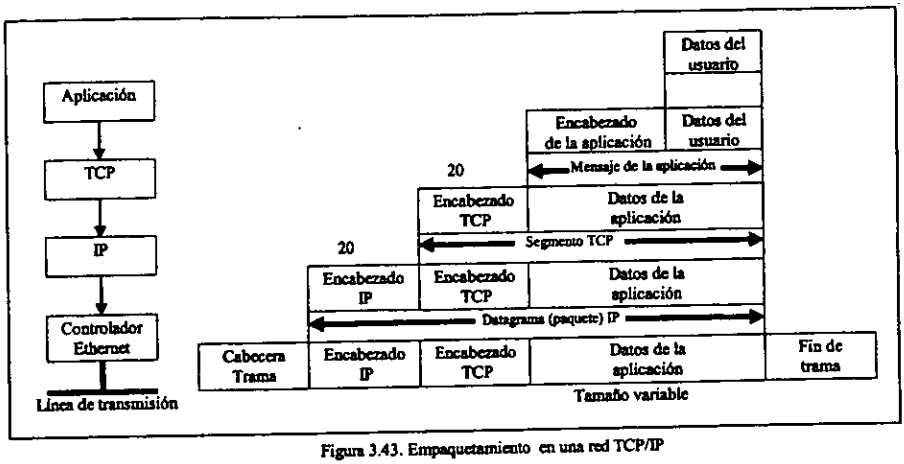


Figura 3.43. Empaquetamiento en una red TCP/IP

3.5.1. Protocolo IP

En la capa de red de TCP/IP, el protocolo IP empaqueta los datos recibidos en *datagramas*. Un datagrama es un paquete de datos independiente de cualquier otro. La transmisión de datagramas es realizada por protocolos que no establecen una conexión previa con la unidad receptora; además, su secuencia de llegada es aleatoria, por lo que le corresponde a la aplicación receptora ordenarlos. Por tanto, IP emplea datagramas no confiables sin conexión para llevar la información a través de la red.

Un datagrama IP consiste en una parte de cabecera y una parte de texto. La cabecera mostrada en la figura 3.44, tienen una parte fija de 20 bytes y una parte opcional de longitud variable.

Los primeros cuatro bits del encabezado IP identifican la versión del protocolo que se utilizó para crear el datagrama. Cuando cambia el formato de los datos de IP, TCP/IP incrementa el número de versión de IP almacenado en el campo *Versión*. Si el software de IP no soporta la versión especificada en el campo *Versión* de cada datagrama, éstos son rechazados asegurando que el software de red no malinterprete los datos contenidos en ellos.

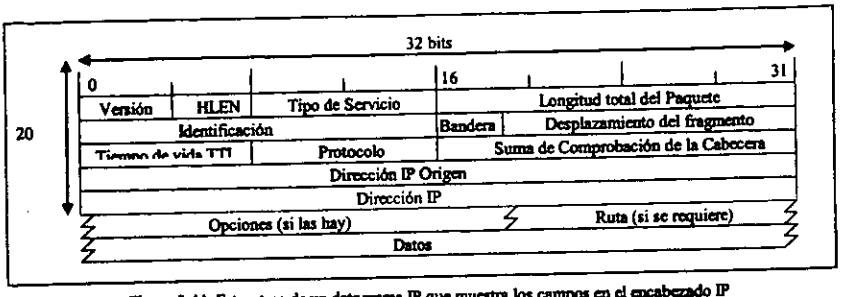


Figura 3.44. Estructura de un datagrama IP que muestra los campos en el encabezado IP

El campo *Longitud de Encabezado* (HLEN, Header Length) especifica el tamaño del encabezado IP en palabras de 32 bits. El valor mínimo es de 5 palabras, cifra que aplica cuando no hay opciones. El valor máximo de este campo es de 15 palabras, lo que limita a la cabecera a 60 bytes y al campo *Opciones* a 40 bytes.

En la figura 3.45 se muestra el campo *Tipo de Servicio* (TOS), el cual define las prioridades del paquete IP. Este campo está dividido en cinco subcampos.

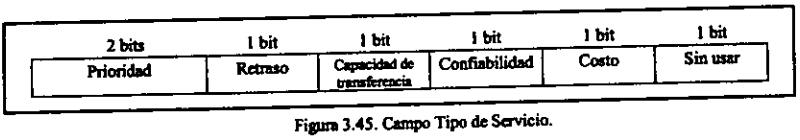


Figura 3.45. Campo Tipo de Servicio.

Los primeros tres bits representan el subcampo prioridad del datagrama. Dentro de él, las aplicaciones o protocolos utilizan un valor dentro del rango de 0 a 7 bits para especificar la importancia de su información. Los siguientes tres bits definen el tipo de transporte deseado para el datagrama. Cuando está activado con uno el bit *Retraso* solicita procesamiento con retardos cortos, el bit *Capacidad de Transferencia* solicita un alto desempeño y el bit *Confiabilidad* solicita alta confiabilidad. Estos bits se emplean en los ruteadores. Por ejemplo, si un ruteador no conoce más que una posible ruta para alcanzar un destino determinado, puede utilizar el campo tipo de transporte para seleccionar una con las características más cercanas a la petición. Por último, el bit *Costo* comunica a la capa de red que desea minimizar el costo.

El campo *Longitud Total del Paquete* define el tamaño del datagrama (incluye tanto la cabecera como los datos). Este campo es de 16 bits, por lo que el tamaño máximo de un datagrama es de 65,535 bytes. El tamaño de los paquetes que se transmiten en una red depende de la tecnología que se emplee. La limitación de paquetes es denomina Unidad de Transferencia Máxima (MTU, Maximum Transfer Unit) de la red. Si una aplicación intenta transmitir un paquete IP más grande que la MTU física de la red, ocurre una fragmentación. IP calcula el punto de inicio de cada fragmento a partir del principio del datagrama y crea cada fragmento en múltiplos de ocho bytes. Después almacena el punto de fragmentación en el campo *Desplazamiento de fragmento* del nuevo datagrama IP.

Puesto que los datagramas suelen dividirse en pequeños fragmentos, es necesario identificarlos mediante un número que está incluido en el campo *Identificación* del encabezado IP. Los anfitriones utilizan este campo de 16 bits para reconocer cada datagrama que envían. Cuando una computadora anfitrión recibe datagramas utiliza el campo *Identificación* para determinar que fragmentos pertenecen a un datagrama.

En el campo *Banderas*, el primer bit está sin uso. El siguiente bit de este campo es el DF (Don't Fragment), que significa no fragmentar, es una indicación para los ruteadores de no fragmentar el datagrama, porque el destino no es capaz de reconstruir los datagramas.

El tercer bit del campo *Banderas* es conocido como MF (More Fragments, Más Fragmentos) . Todos los fragmentos excepto el último tienen establecido este bit, que es necesario para saber cuándo han llegado todos los fragmentos de un datagrama.

El campo *Desplazamiento del fragmento* indica en qué parte del datagrama actual este fragmentado. Dado que existen 13 bits en este campo, puede haber un máximo de 8192 fragmentos por datagrama, dando una longitud máxima de datagrama de 65,536 bytes.

Para llevar a cabo el reensamble de paquetes, el anfitrión destino debe recibir un paquete IP con la bandera MF (More fragments) establecida en uno, para activar un cronómetro de reensamble. Si el cronómetro de reensamble se detiene antes de que el anfitrión reciba todos los fragmentos, éstos se van descartando a medida que lleguen y no se procesa el datagrama. Conforme el anfitrión recolecta los fragmentos en su buffer de reensamble, la capa de red utiliza los campos *Identificación* y *Dirección IP Origen* para determinar que paquetes van juntos. Cuando el anfitrión recibe un fragmento con el bit de la bandera MF en cero, IP calcula la longitud del datagrama original. A partir del fragmento final, en la capa de red, IP calcula la longitud del datagrama original sumando los valores de los campos *Desplazamiento de Fragmento* y *Longitud de Paquete*. Después, IP examina todos los fragmentos relacionados y emplea el campo *Desplazamiento de Fragmento* para volver a combinar los fragmentos en la secuencia adecuada, reensamblando el datagrama original.

El campo *Tiempo de Vida* (TTL, Time to Live), es un contador que sirve para limitar la vida de un paquete. Este contador cuenta el tiempo en segundos (siendo la vida máxima de 255 seg.) y va disminuyendo en cada salto. Cuando el contador llega a cero, el paquete se descarta y se envía de regreso un paquete de aviso al anfitrión de origen. Esta medida evita que los datagramas vaguen eternamente.

Mediante el campo *Protocolo* se indica la capa de transporte a la que debe entregarse. Si este campo contiene el valor 6 el área de datos se empaquetó como un segmento TCP, pero si contiene el valor 17 el software de red empaquetó el área de datos como un datagrama UDP. También existen otros protocolos que usan IP:

Protocolo	Decimal	Binario
ICMP. Protocolo de Control de Mensaje	1	0000001
IGMP. Protocolo de Manejo de Grupos de Internet	2	0000010
TCP. Protocolo de Control de Transporte	6	0000110
UDP Protocolo de Datagrama de Usuario	17	00010001

Tabla 3.6. Valores de los campos de protocolo para el encabezado IP

La *Suma de Comprobación de la Cabecera* verifica solamente la cabecera. Tal suma de comprobación es útil para la detección de errores generados por palabras de memoria erróneas en un ruteador. El algoritmo es sumar todas las medias palabras de 16 bits a medida que llegan, usando aritmética de complemento a uno, y luego obtener el complemento a uno del resultado. Para este algoritmo, la suma de comprobación de la cabecera es cero cuando llega. La suma de comprobación de la cabecera debe volver a calcularse en cada salto, pues cuando menos uno de los campos siempre cambia (como el campo Tiempo de Vida). IP es un protocolo no confiable, por lo que no garantiza la entrega de datos. Sin embargo, mediante la suma de comprobación del encabezado, IP detecta y descarta cualquier paquete corrompido.

El campo *Dirección IP de Origen*, de 32 bits, contiene la dirección IP del anfitrión emisor. Sin importar por cuántos ruteadores pase el paquete en la trayectoria hacia su destino, este campo nunca cambia. Al igual que el campo dirección IP del Origen, el campo *Dirección IP del Destino* contiene una dirección IP estándar de 32 bits. Dependiendo del tipo de mensaje este campo puede contener una dirección IP de un anfitrión o sólo unos para un mensaje de difusión.

El campo *Opciones* se incluye para pruebas de error o de depuración. Este campo no se requiere en todos los datagramas y su longitud varía dependiendo de la opción seleccionada. A su vez, contiene tres subcampos: *Copia*, *tipo de Opción* y *Número de Opción*, los cuales se muestran en la figura 3.46.

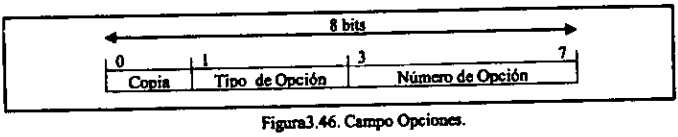


Figura3.46. Campo Opciones.

El bit *Copia* especifica cómo deben manejar los ruteadores las opciones si ocurre la fragmentación de mensajes. Cuando este bit se establece en 1, el ruteador debe copiar las opciones sólo en cada fragmento; cuando se establece en cero, el ruteador debe copiarlas solamente en el primer fragmento.

Los bits del subcampo *Tipo de Opción* especifican una de las clases siguientes:

Tipo de opción	Patrón de bit	Propósito de la opción
0	00	Datagrama o control de red
1	01	Reservado
2	10	Depuración y medición
3	11	Reservado

Tabla 3.7. Tipos de opción en el campo Opciones IP

Dentro de cada clase de opciones, el subcampo Número de Opción selecciona una opción específica de las que se muestras en la tabla3.8.

Tipo de opción	Número de opción	Longitud	Descripción
0	2	11	Seguridad. Especifica que tan secreto es el datagrama.
0	7	Variable	Registro de ruta. Hace que cada ruteador agregue su dirección de IP.
0	3	Variable	Ruteo estricto desde el origen. Indica la trayectoria completa a seguir.
0	9	Variable	Ruteo libre desde el origen. Da una lista de los ruteadores que no deben evitarse
2	4	Variable	Registro de Tiempo. Hace que cada ruteador agregue su dirección y su marca de tiempo.

Tabla 3. 8.Configuraciones para el campo Opciones.

Tales opciones asignan la seguridad y el manejo de restricciones para aplicaciones militares. La opción de *Seguridad* indica que tan secreta es la información. Por ejemplo, un ruteador militar puede usar este campo para especificar que no se encamine a través de ciertos países.

La opción de *Ruteo estricto desde el origen* da la trayectoria completa desde el origen hasta el destino. Se requiere que el datagrama siga esa ruta exacta. Esta opción se usa sobre todo cuando los administradores de sistemas envían paquetes de emergencia porque las tablas de ruteo se han corrompido, o para hacer mediciones de tiempo. La opción de *Ruteo libre desde el origen* requiere que el paquete pase por los ruteadores indicados en la lista, y en el orden especificado, pero se le permite pasar a través de otros ruteadores en el camino.

La opción de *Registrar ruta* indica a los ruteadores a lo largo de la trayectoria que agreguen su dirección de IP al campo de opción. Esto permite a los administradores del sistema buscar fallas en los algoritmos de ruteo.

Por último, la opción *Registro de tiempo* es como la opción registro de ruta, excepto que además de registrar su dirección IP de 32 bits, cada ruteador también registra una marca de tiempo de 32 bits. Usada para la búsqueda de fallas en los algoritmos de ruteo.

Direcciones IP

Cada host y ruteador de Internet tiene una dirección de IP, que codifica su número de red y su número de host. La combinación es única, es decir, no existen dos máquinas que tengan la misma dirección IP. Los números de red los asigna el InterNIC (Inter Network Information Center, Centro de Información de Redes de Internet). De acuerdo a su diseño original, el byte de mayor orden en una dirección IP identifica el número de la red, y los tres bytes de menor orden identifican a la computadora anfitrión. Para este esquema sólo se podían conectar 255 redes, por lo que se ideó otra forma de codificación: la dirección IP usarían los primeros bits del primer byte para identificar una clase de dirección. Las clases de direcciones IP se dividen en A, B, C, D y E. Observe la figura 3.47.

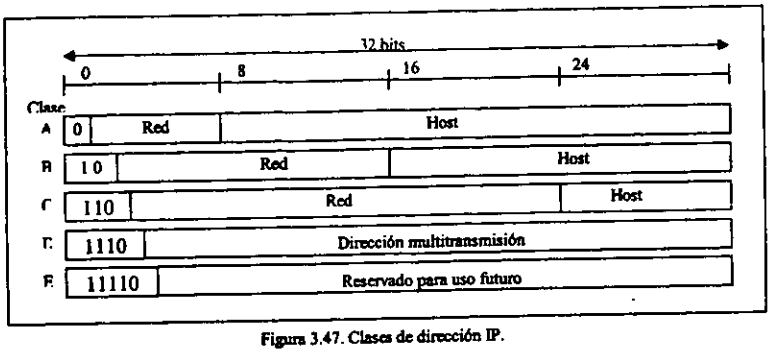


Figura 3.47. Clases de dirección IP.

Una dirección clase A emplea un byte para la identificación del tipo de clase y de red, la cual deja tres bytes libres para el número de identificación del anfitrión. Esto significa que pueden interconectarse únicamente 127 redes con direcciones A (7 bits representa 128 valores pero el 0 es una dirección reservada), y cada una de ellas puede tener 16 777 216 anfitriones.

Las direcciones clase B ocupan un máximo de dos bytes para la identificación del tipo de clase y de red, por lo que deja 16 bits para el número de identificación del anfitrión. De esta clase, tan sólo se pueden obtener 16,384 redes con direcciones de clase B y en cada una de ellas puede existir hasta 65,536 anfitriones.

Las direcciones clase C emplean un máximo de tres bytes para la identificación del tipo de clase y de red. Para el número de identificación del anfitrión se utilizan los 8 bits restantes. De lo anterior, se obtiene 2 097 152 redes con dirección clase C con 255 anfitriones en cada una de ellas.

InterNIC utiliza la clase D para direcciones de transmisión múltiple (multicast). Estas representan a un grupo de computadoras anfitrión en Internet; por lo tanto, la transmisión múltiple envía mensajes a uno o más computadoras anfitrión. InterNIC reserva las direcciones clase E para usos futuros.

Existen direcciones IP reservadas que tienen una función específica. La dirección IP 0.0.0.0 es usada por los anfitriones cuando están siendo arrancados, pero no se usa después. Las direcciones de IP con 0 como números de red se refieren a la red actual. Estas direcciones permiten que las máquinas se refieran a su propia red sin saber su número (pero tienen que saber su clase para saber cuántos ceros hay que incluir). La dirección que consiste solamente en unos permite la difusión en la red local. Las direcciones con un número de red propio y solamente unos en el campo de host permite que las máquinas envíen paquetes de difusión a LAN distantes desde cualquier parte de Internet. Por último, todas las direcciones de la forma 127.xx.yy.zz se reservan para pruebas de realimentación. Los paquetes enviados a esa dirección se procesan localmente y se tratan como paquetes de entrada. Esto permite que los paquetes se envíen a la red local sin que el transmisor conozca su número. Esta característica también se usa para la detección de fallas en el software de la red.

Un problema que surge en una red local a medida que crece es la falta de direcciones disponibles. Como se sabe, InterNIC asigna los número de identificación de red y los administradores asignan el número del anfitrión, por lo que

una solución que han llevado a cabo los administradores es subdividir el espacio de direcciones de anfitrión para crear una red local de redes (subredes). Una dirección de subred es cualquier dirección derivada del esquema de subdivisión de red, el cual sólo cobra significada dentro de la red donde se le definió.

Un ejemplo de ello es la siguiente situación. En la figura 3.48 se tiene una dirección Clase B, en la cual un administrador tiene 16 bits disponibles para los números de identificación del anfitrión. Si estos 16 bits se dividen en dos bytes, uno de ellos puede servir para el número de identificación de la red y el otro como número de identificación del anfitrión. El resultado es una subred de 254 redes conectadas, cada una con 254 anfitriones.

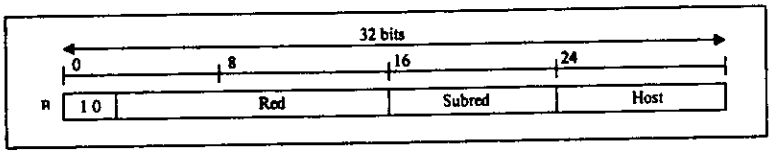


Figura 3. 48.Creación de subredes.

Ruteo IP

Cada ruteador tiene una tabla en la que se lista cierto número de direcciones IP (redes distantes y redes locales). Las tablas de ruteo se basan en que cada anfitrión de la misma red física usa el mismo número de identificación de red, por lo que sólo utilizan los números de identificación de red.

Cada entrada de la tabla de ruteo incluye tres campos: *Red*, *Compuerta* y *Banderas*. Los primeros dos contienen número de identificación de red; el tercero identifica las redes que se enlazan directamente con el propietario de la tabla de ruteo.

El campo *Red* contienen una lista de números de identificación de red. Mientras que el campo *Compuerta* identifica a un ruteador situado en una red especificada en el campo Red. Quizás el ruteador no conecte directamente con la red destinataria, pero mediante las tablas de ruteo se muestra el siguiente salto en la trayectoria hacia un destino específico.

Cuando una computadora anfitrión recibe un paquete, la capa de red extrae el número de red destino del encabezado IP. Luego consulta una tabla de ruteo. Cuando encuentra una entrada para el número de red destino examina el campo Banderas. Si éste indica una conexión directa significa que el paquete es enviado directamente a su destino. Pero si la dirección no está conectada directamente a la red destino se reenvía al siguiente ruteador que se encuentra en la tabla.

3.5.2. Protocolo de Resolución de Direcciones (ARP).

Las direcciones IP no se pueden utilizar para la transmisión de paquetes debido a que el hardware de la capa de enlace de datos no entiende las direcciones IP. Cada tecnología de hardware de red define un mecanismo de direccionamiento que las computadoras utilizan para especificar el destino de cada paquete y solamente se pueden comunicar si conocen sus direcciones físicas de red. De tal manera que cuando un paquete es enviado a través de la red a su destino, la computadora que envíe el paquete tiene que transformar la dirección IP de destino final en su dirección física. Considerando además que, en cualquier punto del camino entre la fuente al destino, el paquete se debe enviar hacia un ruteador intermedio. Por lo tanto, el transmisor tiene que transformar la dirección Internet del ruteador en una dirección física.

Una solución que se encontró para la situación mencionada anteriormente fue el diseño del Protocolo de Asociación de Direcciones (ARP). De manera funcional, ARP está dividido en dos partes: la primer parte transforma una dirección IP en una dirección física cuando se envía un paquete y la segunda responde solicitudes de otras máquinas.

La transformación de la dirección IP a dirección física se realiza de la siguiente manera. Cuando un anfitrión A quiere definir la dirección IP de B, transmite por difusión un paquete especial que pide al anfitrión B que responda con su dirección física. Todos los anfitriones reciben la solicitud, pero sólo el anfitrión B reconoce su propia dirección IP y envía una respuesta que contiene su dirección física. Cuando A recibe la respuesta, utiliza la dirección física para enviar el paquete IP directamente a B. Por lo tanto, el Protocolo de Asociación de direcciones ARP permite que un anfitrión encuentre la dirección física de otro anfitrión dentro de la misma red física con sólo proporcionar la dirección IP de su objetivo.

La difusión es demasiado cara para utilizarse cada vez que una máquina necesita transmitir un paquete a otra, debido a que requiere que cada máquina en la red procese dicho paquete. Para reducir los costos de comunicaciones, las computadoras que utilizan ARP, mantienen una memoria de las asignaciones de direcciones IP a dirección física recientemente adquiridas para que no tengan que utilizar ARP varias veces.

Otra medida para reducir el costo en la transmisión es incluir en la difusión ARP que realiza un transmisor, su asignación de dirección IP y su dirección física; de tal manera que los receptores graben en su memoria intermedia estas direcciones, ya que probablemente serán utilizadas para el envío de respuestas.

En la segunda parte del protocolo, cuando llega un paquete ARP, el software extrae la dirección IP del transmisor y la dirección del hardware, después examina la memoria temporal local para verificar si ya existe un registro para el transmisor. Si es así, el controlador actualiza el registro al sobreescribir la dirección física con la dirección del paquete. Después el receptor procesa el resto del paquete.

Entre el tiempo en que una máquina transmite por difusión su solicitud ARP y recibe la respuesta, los programas de aplicación o los protocolos de un nivel más alto pueden generar solicitudes adicionales para la misma dirección; por lo tanto, el software debe de recordar que ya envió una solicitud para no enviar más. Por lo común, el software ARP

coloca los paquetes adicionales en una cola de espera. Una vez que llega la respuesta y se conoce la asignación de dirección, el software ARP remueve los paquetes de la cola de espera, pone cada paquete en una trama y utiliza la asignación de dirección para llenar la dirección física del destino.

3.5.3. Protocolo de Resolución de Direcciones en Reversa (RARP).

El Protocolo de Resolución de Direcciones Inverso mapea una dirección física, en una dirección IP. Los desarrolladores de TCP/IP diseñaron RARP para que lo usaran en computadoras sin disco duro. Por ejemplo, una estación de trabajo sin disco duro puede leer la dirección de su capa de enlace de su tarjeta de interfaz de red.
Con RARP, una estación de trabajo sin disco duro puede difundir una solicitud que pide a otro anfitrión de la red que revise la dirección física y después le informe la dirección IP correcta. Al usar su propia dirección IP, la estación de trabajo puede transmitir un mensaje pidiendo a otro sistema que le cargue el sistema operativo. Por tanto, se puede enlazar estaciones de trabajo sin disco duro a Internet para después arrancar el sistema desde servidores remotos de la red.
Al igual que un mensaje ARP, un mensaje RARP se envía de una máquina a otra, encapsulado en la porción de datos de una trama de red. El que envía transmite por difusión una solicitud RARP especificada como máquina transmisora y receptora, y proporciona su dirección física de red en el campo de direcciones de hardware objetivo. Todas las máquinas en la red reciben la solicitud, pero sólo las autorizadas para proporcionar el servicio de RARP la procesan y envían la respuesta; dichas máquinas se conocen de manera informal como servidores RARP. Para que RARP funcione correctamente, la red debe contener por lo menos un servidor RARP.
Una vez llenado el campo de dirección de protocolo objetivo, los servidores contestan las solicitudes, cambian el tipo de mensaje de solicitud a respuesta y envían ésta de vuelta directamente a la máquina que la solicitó. La máquina original recibe respuestas de todos los servidores RARP, aunque sólo se necesite una contestación.

3.5.4. Protocolo de Control de Mensaje de Internet (ICMP).

Para permitir que los ruteadores en una red de redes 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 Protocolo de Mensajes de Control Internet (ICMP), se considera como parte obligatoria del IP. Este mecanismo de reporte de errores proporciona a los ruteadores la forma de reportar a la fuente original el error que encuentren. Sin embargo, aunque ICMP sugiere acciones posibles para responder a los reportes de error, no especifica del todo la acción que debe tomarse para cada posible error. Por tanto, cuando un datagrama causa un error, el ICMP sólo puede reportar la condición del error a la fuente original del

datagrama; correspondiéndole a la fuente relacionar el error con un programa de aplicación individual o tomar alguna otra acción para corregir el problema.

Los mensajes ICMP requieren dos niveles de encapsulación como se muestra en la figura 3.49. Cada mensaje ICMP viaja a través de la red de redes en la porción de datos de un datagrama IP, el cual viaja a través de cada red física en la porción de una trama. Los datagramas que llevan mensajes ICMP se encaminan exactamente como los que llevan información de usuario; no existe ni una confiabilidad ni una prioridad adicionales. Por lo tanto, los mensajes de error se pueden perder o descartar.

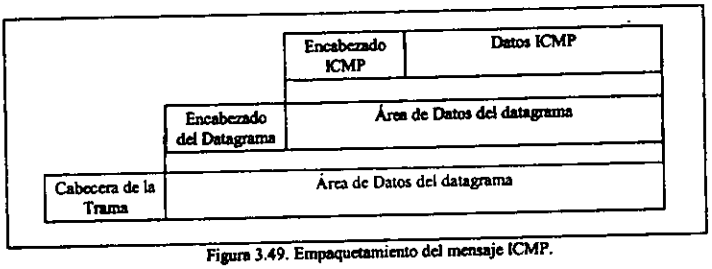


Figura 3.49. Empaquetamiento del mensaje ICMP.

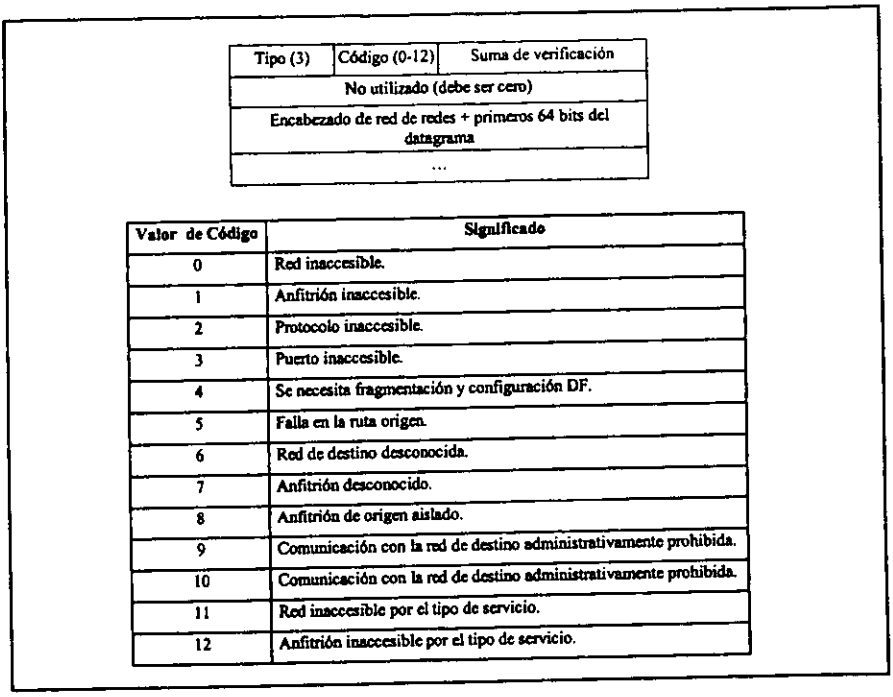
Una situación que puede presentarse en un datagrama IP que lleva un mensaje ICMP es que este cause un error. Para evitar el problema de tener mensajes de error sobre error, se ha estipulado que los mensajes ICMP no se generan por errores resultantes de datagramas que llevan mensajes de error ICMP.

Aunque cada mensaje ICMP tienen su propio formato, todos comienzan con los mismos tres campos: *Tipo de mensaje*, de 8 bits y números enteros, que identifica el mensaje; un campo *Código* de 8 bits, que proporciona más información sobre el tipo de mensaje, y un campo *Suma de verificación*, de 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. Se han definido una docena de tipos de mensaje de ICMP; los más importantes se listan en la tabla 3.9. Cada tipo de mensaje se empaqueta en un paquete IP.

Tipo de mensaje	Descripción
Destino inalcanzable	No pudo entregarse el paquete
Tiempo excedido	Campo de tiempo de vida llegó a cero
Problema de parámetro	Campo de cabecera no válido
Supresión de origen	Paquete de estrangulamiento
Reenvío	Enseña geografía a un ruteador
Solicitud de eco	Pregunta a una máquina si está viva
Respuesta de eco	si, estoy viva
Solicitud de marca de tiempo	Igual que la solicitud de eco, pero con marca de tiempo
respuesta de marca de tiempo	Igual que la respuesta de eco, pero con marca de tiempo

Tabla3.9. Los principales tipos de mensajes ICMP.

El mensaje *destino inalcanzable* (destination unreachable), mostrado en la figura 3.50, se usa cuando la subred o un ruteador no pueden ubicar el destino o cuando un ruteador necesita fragmentar un datagrama pero está activado el bit DF (no fragmentar). El campo *Código* de un mensaje de destino inalcanzable contiene los siguientes posibles valores enteros:



Tipo (3)	Código (0-12)	Suma de verificación
No utilizado (debe ser cero)		
Encabezado de red de redes + primeros 64 bits del datagrama		
...		

Valor de Código	Significado
0	Red inaccesible.
1	Anfitrión inaccesible.
2	Protocolo inaccesible.
3	Puerto inaccesible.
4	Se necesita fragmentación y configuración DF.
5	Falla en la ruta origen.
6	Red de destino desconocida.
7	Anfitrión desconocido.
8	Anfitrión de origen aislado.
9	Comunicación con la red de destino administrativamente prohibida.
10	Comunicación con la red de destino administrativamente prohibida.
11	Red inaccesible por el tipo de servicio.
12	Anfitrión inaccesible por el tipo de servicio.

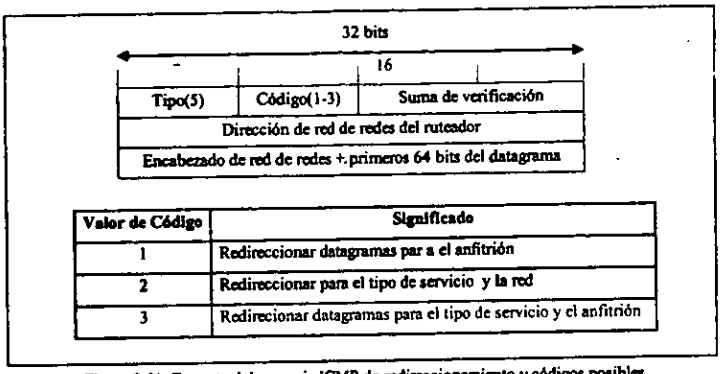
Figura 3.50. Formato del mensaje ICMP de destino inaccesible.

El mensaje de *tiempo excedido* (time exceeded) se envía cuando un paquete se descarta debido a que su contador llega a cero. Este suceso es un síntoma de que los paquetes están en ciclo, de que hay un congestionamiento enorme, o de que los valores de temporización son demasiados bajos.

El mensaje de *problema de parámetro* (parameter problem) indica que ha detectado un valor ilegal en un campo de cabecera. Este problema indica una falla en el software de IP del anfitrión, o posiblemente en el software de un ruteador transitado.

El mensaje *supresión de origen* (source quench) es una solicitud para que la fuente reduzca la velocidadd de transmisión de datagramas. Por lo general, los ruteadores congestionados envían un mensaje de *supresión de origen* por cada datagrama que descartan.

El mensaje de *redireccionamiento* (redirect) se usa cuando un ruteador se da cuenta de que un paquete parece estar mal encaminado, reportando al anfitrión transmisor el posible error. El formato del mensaje de *redireccionamiento* se muestra en la figura 3.51.



Valor de Código	Significado
1	Redireccionar datagramas par a el anfitrión
2	Redireccionar para el tipo de servicio y la red
3	Redirecionar datagramas para el tipo de servicio y el anfitrión

Figura 3.51. Formato del mensaje ICMP de redireccionamiento y códigos posibles.

El campo *Dirección* de red de redes contiene la dirección de un ruteador que el anfitrión utilizará para alcanzar el destino mencionado en el encabezado del datagrama. El campo *Encabezado de red de redes* contiene el encabezado IP, más los siguientes 64 bits del datagrama que activó el mensaje. Por lo tanto, un anfitrión que recibe un redireccionamiento ICMP examina el prefijo del datagrama para determinar la dirección destino. El campo *Código* de un mensaje ICMP de redireccionamiento especifica con mayor detalle cómo interpretar la dirección destino.

Los mensajes de *solicitud de eco* (echo request) y *respuesta de echo* (echo replay) sirven para ver si un destino dado es alcanzable y está vivo. Al recibir el mensaje de eco, se espera que el destino devuelva un mensaje de respuesta de eco. El formato de este mensaje se muestra en la figura 3.52.

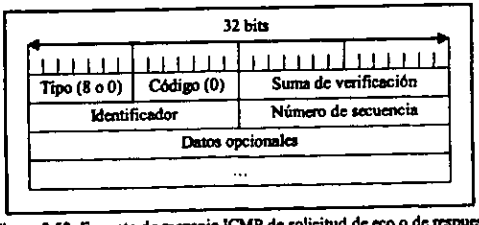


Figura 3.52. Formato de mensaje ICMP de solicitud de eco o de respuesta

El campo *Datos opcionales* tiene una longitud variable que contiene los datos que se regresarán al transmisor. Una respuesta de eco siempre regresa exactamente los mismos datos que se recibieron en la solicitud. Los campos *Identificador* y *Número de secuencia* los utiliza el transmisor para responder a las solicitudes. El valor del campo *Tipo* especifica si el mensaje es una solicitud o una respuesta.

Los mensajes *solicitud de marca* de tiempo (timestamp request) y *respuesta de marca* de tiempo (timestamp reply) son parecidos a los mensajes de eco, excepto que el tiempo de llegada del mensaje y el tiempo de partida de la respuesta se registran en la respuesta. Este recurso se emplea para medir el desempeño de la red. En la siguiente figura se muestra el formato del mensaje de solicitud de marca de tiempo.

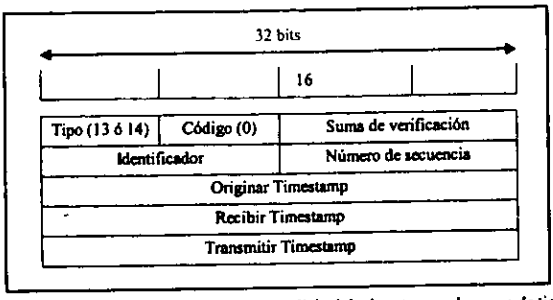


Figura 3.53. Formato del mensaje ICMP de solicitud de timestamp o de marca de tiempo

El campo *Tipo* identifica el mensaje como solicitud (13) o como respuesta (14); los campos *Identificador* y *Número de secuencia* los utiliza la fuente para asociar las solicitudes con las respuestas. Los campos restantes *Originar Timestamp* es llenado por la fuente original justo antes de transmitir el paquete, el campo *Recibir Timestamp* es llenado al recibir una solicitud y el campo *Transmitir Timestamp* se llena justo antes de transmitir la respuesta.
Los anfitriones utilizan estos tres campos para calcular estimaciones del tiempo de retraso entre ellos y para sincronizar sus relojes. Debido a que la respuesta incluye el campo *Originar Timestamp*, un anfitrión puede calcular el tiempo requerido para que una solicitud viaje hasta un destino, se transforme en una respuesta y regrese.

3.5.5 Protocolo de Manejo de Grupos de Internet

Las direcciones IP se dividen en tres categorías: transmisión unitaria, difusión y transmisión múltiple. Las direcciones clase A, B y C son de transmisión unitaria, ya que identifican a un solo anfitrión. Una dirección de difusión especifica que los conmutadores de paquetes encaminan los datos a todos los anfitriones de la red.
Una dirección de transmisión múltiple identifica a un grupo de anfitriones específico en Internet. La pertenencia a un grupo de multidifusión es un proceso dinámico, ya que una computadora anfitrión puede unirse o retirarse de él en cualquier momento.
Cada grupo de multidifusión tiene una dirección de multidifusión única, algunas de estas direcciones son asignadas por la autoridad de Internet y corresponden a grupos que siempre existen aun cuando no tengan miembros. Estas direcciones se dice que *son bien conocidas*. Otras direcciones de multidifusión están disponibles para usos

temporales. Corresponde a *grupos transitorios de multidifusión* que se crean cuando son necesarios y se deshecha cuando el número de miembros llega a cero.

Recordando el formato de direcciones de multidifusión IP, los primeros cuatro bits hace referencia al tipo de dirección, que para este caso es de clase D (multidifusión) y los siguiente 28 bits identifican al grupo. Ver figura 5.54.

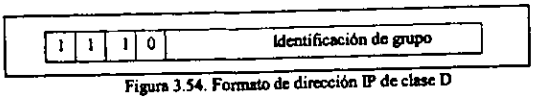


Figura 3.54. Formato de dirección IP de clase D

La multidifusión IP a través de una red de redes, requiere de ruteadores especiales de multidifusión para el envio de datagramas. Los ruteadores de multidifusión utilizan las capacidades de multidifusión de hardware local para entregar el datagrama en la red o las redes de destino que soporten la multidifusión. Para transformar una dirección de multidifusción IP en multidifusión de hardware local, mencionando como caso especial Ethernet, se requiere colocar los 23 bits de orden menor de la dirección de multidifusión IP dentro de los 23 bits de orden inferior de la dirección de multidifusión Ethernet especial $01.00.5E.00.00_{16}$. Por ejemplo, la dirección de multidifusión IP 224.0.0.1 se convierte en la dirección de multidifusión Ethernet $01.00.5E.00.01_{16}$.

Implementación IGMP

Para participar en la multidifusión IP dentro de la una red local, un anfitrión debe tener el software que le permita enviar y recibir datagramas de multidifusión. Para participar en una multidifusión que cubra varias redes, el anfitrión debe informar a los ruteadores de multidifusión local. El ruteador local se pone en contacto con otros ruteadores de multidifusión, pasando información hacia los miembros y estableciendo rutas. Antes de que un ruteador de multidifusión pueda difundir información los miembros de multidifusión, debe determinar si uno o más anfitriones en la red local han decidido unirse a un grupo de multidifusión. Para hacerlo, lo ruteadores de multidifusión y los anfitriones que implementa la multidifusión deben utilizar el Protocolo de Manejo de Grupos de Internet IGMP para comunicar información a los miembros del grupo.

El IGMP tiene dos fases, la primera cuando un anfitrión se une a un nuevo grupo de multidifusión envía un mensaje IGMP para la dirección de multidifusión "todos los anfitriones", declarando su membresía. Los ruteadores de multidifusión local reciben el mensaje y establecen el camino para difundir la información de membresía del grupo hacia otros ruteadores de multidifusión a través de la red de redes. En la segunda fase, debido a que la membresía es dinámica, los ruteadores de multidifusión local muestrean de manera periódica a los anfitriones en la red local para determinar qué anfitriones se mantiene como miembros de qué grupos. Si en un grupo no se reportan miembros después de varios muestreos, el ruteador de multidifusión asume que no hay anfitriones en la red que se mantengan en el grupo y deja de anunciar miembros del grupo a otros ruteadores de multidifusión.

El IGMP está diseñado para evitar congestionamiento en una red local:

- Toda la comunicación entre anfitriones y ruteadores de multidifusión utilizan multidifusión IP. En las redes en las que el hardware soporta multidifusión, los anfitriones que no participan en la multidifusión IP nunca reciben mensajes IGMP.
- Un ruteador de multidifusión no enviara mensajes de solicitud individuales para cada grupo de multidifusión, sino un mensaje de muestreo para solicitar información relacionada con la membresía en todos los grupos.
- Y, los anfitriones que son miembros de varios grupos no envían respuestas múltiples al mismo tiempo. Luego de que un mensaje de solicitud IGMP llega desde un ruteador de multidifusión, el anfitrión asigna un retardo aleatorio de entre 0 y 10 segundos para cada grupo en el que tiene miembros, y envía una respuesta para este grupo después del retardo.

Transiciones del estado de la membresía de grupo

El IGMP debe recordar el estado de cada grupo de multidifusión al que el anfitrión pertenece mediante una tabla. Inicialmente, todos los espacios en la tabla estarán sin usarse. Cada vez que un programa de aplicación en el anfitrión se una a un nuevo grupo, el software IGMP asignará un espacio y lo llenará con información acerca del grupo. Entre la información, el IGMP establecerá un contador de referencia de grupo, el cual se inicie en 1. Si aplicaciones adicionales se unen al grupo, el IGMP incrementará el contador de referencia en la información almacenada. Conforme los programas de aplicación abandonan el grupo, el IGMP decrementa el contador. El anfitrión deja el grupo de multidifusión cuando el contador llega a cero.

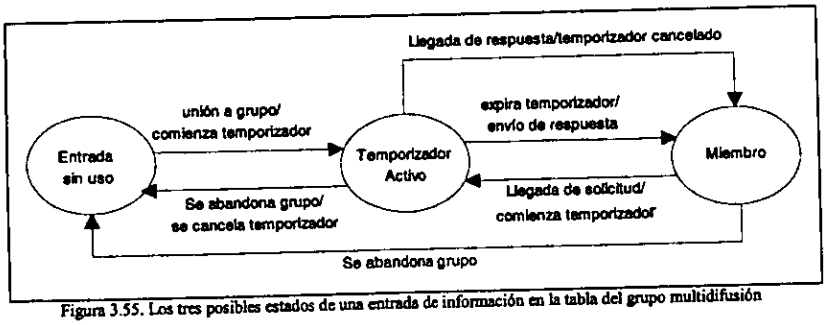


Figura 3.55. Los tres posibles estados de una entrada de información en la tabla del grupo multidifusión

En el esquema de la figura 3.55, una solicitud de unión a un grupo coloca la entrada de información en el estado *Temporizador Activo* y ajusta el temporizador con un valor pequeño. Cuando el temporizador expira, el IGMP genera y envía un mensaje de respuesta y cambia la entrada de información al estado *Miembro*.

En el estado *Miembro*, la recepción de una solicitud IGMP ocasiona que el software elija un valor de tiempo, inicie un temporizador para la entrada de información y cambie la información de entrada al estado *Temporizador Activo*.

Si otros anfitriones envían una respuesta para el grupo de multidifusión, luego de que el temporizador expira, el IGMP cancela el temporizador y cambia la entrada de información de nuevo al estado *Miembro*.

Formato de los mensajes IGMP

Los mensajes IGMP se valen de datagramas IP para transportar mensajes, este protocolo es parte integral del Protocolo Internet. El formato de un mensaje IGMP (Figura 3.56), esta constituido por un campo *Versión*, el cual, contiene la información sobre la versión del protocolo. También contiene un campo *Tipo*, que identifica al mensaje como una solicitud por un ruteador de multidifusión (tipo 1) o como una respuesta enviada por un anfitrión (tipo 2). El campo *Sin uso* debe ser 0 y el campo *Suma de verificación* contiene una suma de comprobación para el mensaje IGMP de 8 bytes.

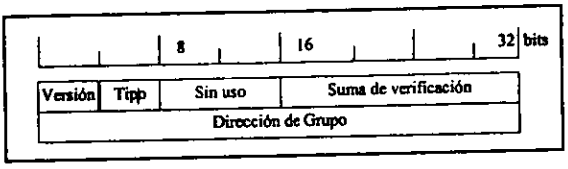


Figura 3.56 Formato de un mensaje IGMP.

Los anfitriones emplean el campo Dirección de grupo para reportar una membresía en un grupo de multidifusión en particular.

3.5.6 Los Protocolos de Transporte de TCP/IP

TCP/IP incluye dos protocolos de transporte: el Protocolo de Control de Transporte TCP y el Protocolo de Datagrama de Usuario UDP. El primero es un protocolo orientado a conexión que utiliza flujo de bytes confiable para enviar y recibir datos. Además proporciona un circuito virtual para las comunicaciones en red. El segundo es un protocolo sin conexión, no confiable, que utiliza datagramas para enviar y recibir datos.

Puertos de la Capa de Transporte

En TCP/IP un puerto es como una dirección IP, excepto que éste es asociado a un protocolo, no a una computadora anfitrión. Del mismo modo que los datagramas IP almacenan las direcciones IP origen y destino, los protocolos de transporte almacenan los números de los puertos del origen y del destino.

3.5.6.1. Protocolo de Datagrama de Usuario (UDP).

El protocolo UDP proporciona el mecanismo para que los programas de aplicación puedan enviar datagramas a otros programas de aplicación. El UDP proporciona puertos de protocolo 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 éste envíe una respuesta.

El UDP utiliza el Protocolo Internet para transportar un mensaje de una máquina a otra y proporciona la misma semántica de entrega de datagramas, sin conexión y no confiable que el IP. No emplea acuses de recibo para asegurarse de que lleguen 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 sin orden. Además, los paquetes pueden llegar más rápido de lo que el receptor los puede procesar.

El protocolo de datagrama de usuario proporciona un servicio de entrega sin conexión y no confiable, utilizando el IP para transportar mensajes entre máquinas. Emplea el IP para llevar mensajes, pero agrega la capacidad para distinguir entre varios destinos dentro de una computadora anfitrión.

Formato de los mensajes 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 UDP. Vea la siguiente figura.

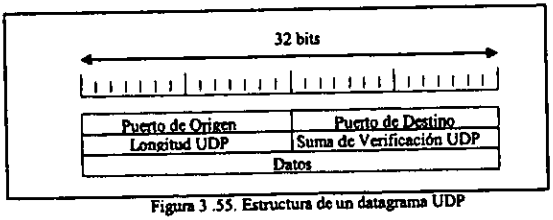


Figura 3 .55. Estructura de un datagrama UDP

La cabecera cuenta con 8 bytes de longitud e incluye cuatro campos: *Puerto de Origen, Puerto de Destino, Longitud y Suma de Verificación*. El campo *Puerto Origen*, cuya longitud es de 8 bits, identifica el número del puerto de protocolo que envían los datos. El campo *Puerto Destino* de 8 bits, identifica el puerto que los recibirá. El campo *Longitud* especifica la longitud del datagrama UDP, incluyendo la cabecera. La suma de verificación es opcional y no es necesario utilizarla; un valor de cero en el campo *Suma de Verificación* significa que la suma no se computó.

La suma de verificación UDP abarca más información de la que está presente en el datagrama UDP. Para realizar la suma de verificación, el UDP añade un pseudocabecera al datagrama UDP y un octeto de ceros para rellenar el datagrama, con el fin de alcanzar exactamente un múltiplo de 16 bits. La suma de verificación se realiza sobre todo el conjunto. El byte utilizado como relleno y la pseudocabecera no se transmiten con el datagrama UDP, ni se incluyen en su longitud. Para llevar a cabo la suma de verificación, el software primero almacena un cero en el campo de *Suma de Verificación*, luego, acumula una suma de complemento de 16 bits de todo el conjunto, incluyendo la pseudocabecera, la cabecera UDP y los datos del usuario. El propósito de utilizar la pseudocabecera es verificar que el datagrama UDP llegó a su destino correcto. Para verificar el destino, el UDP en la máquina transmisora calcula una suma de verificación que cubre tanto la dirección IP de destino como el datagrama UDP. En el destino, el software UDP revisa la suma de verificación utilizando la dirección IP de destino, obtenida de la cabecera del datagrama IP que transportó el mensaje UDP. Si la suma concuerda, debe ser verdad que el datagrama llegó al anfitrión de destino deseado, así como al puerto de protocolo correcto dentro del anfitrión.

La pseudocabecera, mostrada en la figura 3.58, consiste de 12 bytes. Los campos *Dirección IP de Origen, Dirección IP de Destino* , contienen las direcciones IP que se utilizarán cuando se envíe el mensaje UDP. El campo *Proto* contiene el código del tipo de ptrotocolo IP (17 para UDP) y el campo *Longitud UDP* contiene la longitud del datagrama UDP (sin incluir la pseudocabecera).

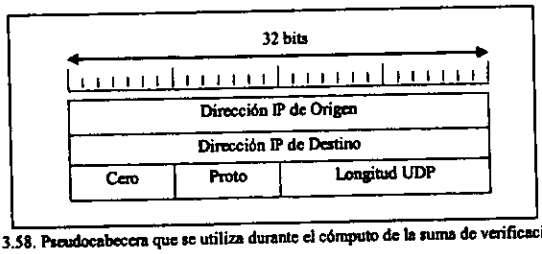


Figura 3.58. Pseudocabecera que se utiliza durante el cómputo de la suma de verificación UDP.

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 él, tendrá el número de puerto en el campo *Puerto de Origen*. UDP acepta los datagramas entrantes y después los ordena y distribuye (demultiplexa) basado en el número de puerto del destino. En la figura 3.59 podemos ver un esquema de recepción de datagramas UDP.

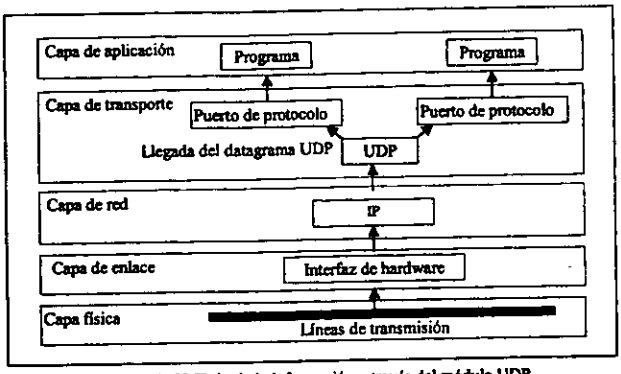


Figura 3. 59.Flujo de la información a través del módulo UDP.

Número de puerto UDP reservados y disponibles

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. Este enfoque consiste en la asignación de puertos conforme un programa de red lo solicite. Para conocer la asignación actual de puerto en otra computadora, es necesario enviar una solicitud preguntando por el puerto que utiliza el servicio de transferencia de archivos, respondiendo la máquina objetivo, el número de puerto correcto a utilizar.

TCP/IP adopta un enfoque híbrido que preasigna algunos números de puerto, pero 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.

Decimal	Palabra clave	Descripción	Decimal	Palabra clave	Descripción
0	-	Reservado	19	CHARGEN	Generador de caracteres
7	ECHO	Eco	37	TIME	Hora
9	DISCARD	Descartar	42	NAMESERVER	Servidor de nombre de anfitriones
11	USERS	Usuarios Archivos	43	NICNAME	Quien es
13	DAYTIME	Hora del día	53	DOMAIN	Servidor de nombres de dominios
15	-	Quién está ahí	67	BOOTPS	Servidor de protocolo bootstrap
17	QOUTE	Cita del día	68	BOOTPC	Cliente de protocolo bootstrap

Tabla 3.10. Puertos UDP asignados

3.5.6.2. Protocolo de Control de Transporte (TCP).

El Protocolo de Control de Transmisión es un protocolo orientado a conexión, el cual proporciona servicios tanto para las capas superiores, como para las inferiores. TCP se diseñó específicamente para proporcionar un flujo de bytes confiable a través de una red no confiable. Si un paquete se corrompe o se pierde, TCP es el que maneja la retransmisión. Además, debe ser capaz de manejar la terminación de una aplicación en una capa superior, que estaba esperando la llegada de datagramas, así como fallas en capas inferiores. Otro servicio que proporciona este protocolo es el de asegurar que las prioridades sean respetadas. El aislamiento de estos servicios en una capa por separado permite que las aplicaciones se diseñen sin preocuparse del control del flujo o la confiabilidad del mensaje.

Características del servicio de entrega confiable

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

- *Orientación de flujo.* Cuando dos programas de aplicación (procesos de usuario) transfieren grandes volúmenes de información, el servicio de entrega de flujo en la máquina destino pasa al receptor exactamente la misma secuencia de bytes que le pasa el transmisor en la máquina origen
- *Conexión de circuito virtual.* Antes de empezar la transferencia de datos, los programas de aplicación, el transmisor y el receptor informan a sus respectivos sistemas operativos la posible realización de una transferencia de flujo. Posteriormente, una aplicación realiza una llamada a la otra y después de que ésta la acepta, los módulos de software del protocolo en los dos sistemas empiezan a enviarse mensajes, verificando que la transferencia esté autorizada y que los dos extremos estén 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 máquinas continúan comunicándose para verificar que los datos se reciban correctamente. Si la comunicación no se logra por cualquier motivo, ambas máquinas detectarán la falla y la reportarán a los programas apropiados de aplicación. La conexión realizada es conocido como circuito virtual, aunque los programas de aplicación visualizan la conexión como un circuito dedicado de hardware.
- *Transferencia con memoria intermedia.* Cuando una aplicación pasa datos al TCP, éste puede enviarlos de inmediato o guardarlos en un buffer (se reúnen datos suficientes de un flujo para llenar un datagrama del tamaño estipulado antes de transmitirlo a través de una red de redes). Cuando la aplicación requiere que los datos se envíen de inmediato el servicio de flujo proporciona un mecanismo de empuje (push) que las aplicaciones utilizan para forzar una transferencia. En el extremo transmisor, un empuje obliga al software de protocolo a transferir

todos los datos generados sin tener que esperar a que se llene una memoria intermedia. Cuando llega al extremo receptor, el empuje hace que el TCP ponga los datos a disposición de la aplicación sin demora.

- *Flujo no estructurado.* El servicio de flujo TCP/IP no está obligado a formar flujos estructurados. No existe forma para que una aplicación haga que un servicio de flujos marque fronteras o identifique los datos que se estén enviando.
- *Conexión Duplex.* Las conexiones proporcionadas por el servicio de flujo TCP/IP permiten la transferencia concurrente en ambas direcciones (duplex). El servicio de flujo puede permitir 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 semiduplex. Pero, la ventaja de una conexión duplex es que el software subyacente de protocolo puede enviar en datagramas información de control de flujo al origen y al mismo tiempo, llevar datos en la dirección opuesta.

Confiabilidad

Para asegurar la confiabilidad y la secuencia del flujo de bytes, los protocolos confiables utilizan la técnica de 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) 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. Este protocolo se visualiza en la figura 3.60.

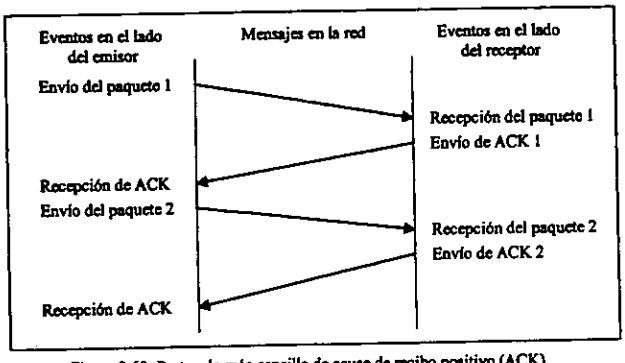


Figura 3.60. Protocolo más sencillo de acuse de recibo positivo (ACK).

El problema final de confiabilidad surge cuando un sistema de entrega de paquetes los duplica (Figura 3.61). Los duplicados también pueden surgir cuando las redes tienen grandes retrasos que provocan la retransmisión prematura.

Por lo general, los protocolos confiables detectan los paquetes duplicados al asignar a cada uno un número de secuencia y al obligar al receptor a recordar qué números de secuencia recibe.

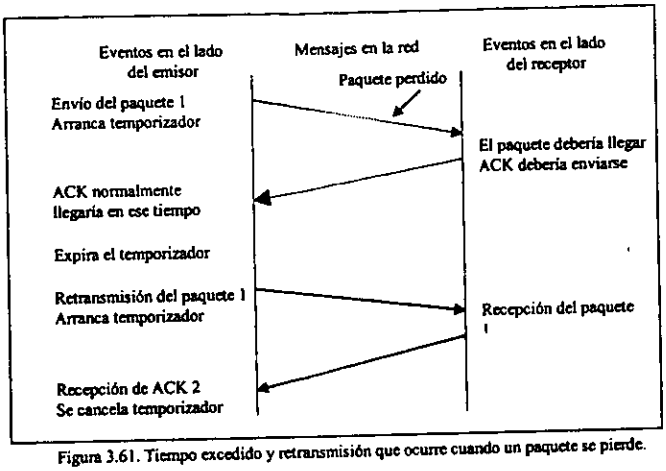


Figura 3.61. Tiempo excedido y retransmisión que ocurre cuando un paquete se pierde.

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, para que el receptor pueda asociar correctamente los acuses de recibo con los paquetes.

Ventanas deslizables

Un protocolo simple de acuses de recibo positivos ocupa un mayor 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. Para mejorar esta capacidad de transferencia de los mensajes, TCP utiliza el concepto de ventana deslizable.
El protocolo coloca una ventana pequeña y de tamaño fijo en la secuencia, y transmite todos los paquetes dentro de la ventana. Vea la siguiente figura.

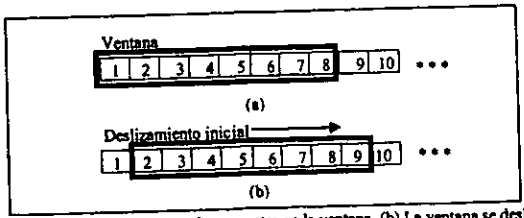


Figura 3.62. (a)Ventana deslizante con ocho paquetes en la ventana. (b) La ventana se desliza al paquete 9

Un paquete es del tipo "acknowledged" o sin acuse de recibo si se transmitió pero no se recibió ningún acuse de recibo. En la figura 3.63, 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.

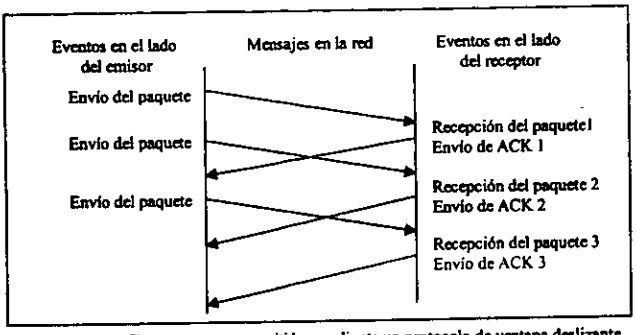


Figura 3. 63. Tres paquetes transmitidos mediante un protocolo de ventana deslizante.

Un protocolo de ventana deslizable siempre recuerda qué paquetes tienen acuse de recibo y mantiene un temporizador separado para cada paquete sin acuse de recibo. Si se pierde un paquete, el temporizador concluye y el transmisor reenvía el paquete. Cuando el receptor desliza su ventana, mueve hacia atrás todos los paquetes con acuse. En el extremo receptor, el software de protocolo mantiene una ventana análoga, que acepta y acusa como recibidos los paquetes conforme llegan. Por lo tanto, la ventana divide la secuencia de paquetes en tres partes: los paquetes a la izquierda de la ventana se transmitieron, recibieron y acusaron exitosamente; los paquetes a la derecha no se han transmitido; y los paquetes que quedan dentro de la ventana están en proceso de transmisión. El paquete con menor número en la ventana es el primer paquete en la secuencia para el que no se ha hecho un acuse de recibo.

El mecanismo TCP de ventana deslizante opera a nivel de byte, no a nivel de paquete. Los bytes del flujo de datos se numeran de manera secuencial, y el transmisor guarda tres apuntadores asociados con cada conexión. Los apuntadores definen una ventana deslizante, la cual se puede observar en la figura 3.64.

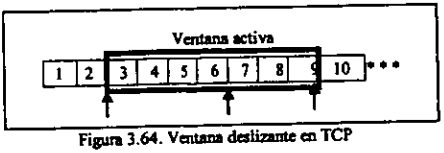


Figura 3.64. Ventana deslizante en TCP

El primer apuntador marca el extremo izquierdo de la ventana deslizante, separa los bytes que ya se enviaron y envía el acuse de recibo de los bytes ya enviados. Un segundo apuntador marca el extremo derecho de la ventana deslizante y define el byte más alto en la secuencia que se puede enviar antes de recibir más 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 bytes dentro de la ventana. Mientras. la frontera de la ventana se mueve rápidamente de izquierda a derecha.

Las conexiones TCP son de tipo duplex, se llevan a cabo dos transferencias al mismo tiempo en cada conexión, una en cada dirección. Por lo tanto, el software TCP en cada extremo mantiene dos ventanas por cada conexión, una se desliza a lo largo del flujo de datos que se envía, mientras la otra se desliza a lo largo de los datos que se recibe.

El TCP permite que el tamaño de la ventana varíe. Cada recibo, que informa cuántos bytes se recibieron, contiene un aviso de ventana que especifica cuántos bytes adicionales de datos está preparado para aceptar el receptor. Si la memoria intermedia del receptor se llena, no puede aceptar más paquetes, así que envía un anuncio de ventana más pequeña. En caso extremo, el receptor anuncia un tamaño de ventana igual a cero para detener toda la transmisión. Después, cuando hay memoria intermedia disponible, el receptor anuncia un tamaño de ventana distinto a cero para activar de nuevo el flujo de datos.

Puertos en TCP

Los puertos que utiliza TCP son diferentes a los de UDP. El protocolo sin conexión UDP entrega los datos al puerto sin mantener una conexión entre el emisor y el receptor. En cambio, en TCP establece una conexión para hacer la entrega de los datos. Una conexión está definida por un par de puntos extremos. En TCP se define a un punto extremo como un par de números enteros (anfitrión, puerto), en donde anfitrión es la dirección IP de un anfitrión y puerto es un puerto TCP en dicho anfitrión. Por ejemplo, el punto extremo (128.19.2.3,25), se refiere al puerto TCP 25 en la máquina con dirección IP 128. 10.2.3. También dos conexiones pueden utilizar al mismo tiempo un puerto TCP, debido a que este protocolo asocia los mensajes entrantes con una conexión en vez de hacerlo con un puerto, utiliza ambos puntos extremos para identificar la conexión apropiada

Cabecera TCP

Una unidad de datos TCP es un segmento. Cada segmento comienza con una cabecera de formato fijo de 20 bytes. La cabecera fija, mostrada en la figura 3.65, puede ir seguida de opciones de cabecera. Tras las opciones, si las hay, puede haber hasta 65,515 bytes de datos.

Los campos Puerto Origen y Puerto Destino, ambos campos de 16 bits, identifican de manera efectiva las aplicaciones enviadas y recibidas. Los números de los campos Puerto Origen y Destino, más la dirección IP origen y destino, se combinan para crear una identificación única de cada conexión TCP. Puede llamar socket a cada lado de conexión TCP.

Los campos *Número de secuencia* y *Número de acuse de recibo* son de 32 bits. El primero identifica el primer byte de datos en el área de datos del segmento TCP. El segundo identifica el siguiente byte de datos que espera recibir la conexión del flujo de datos.

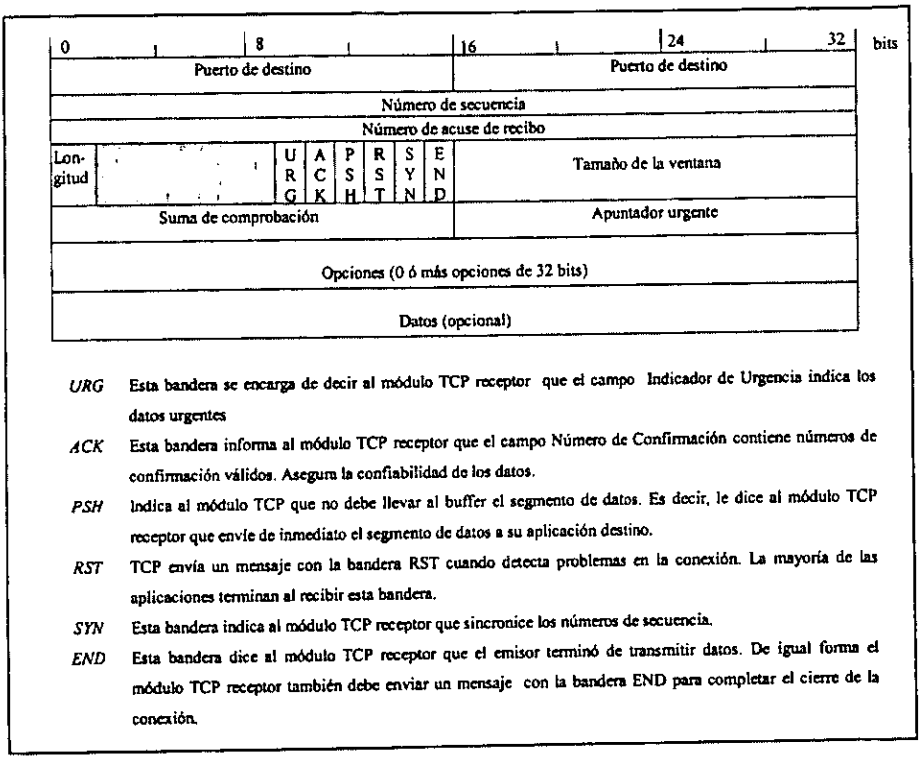


Figura 3.65. Cabecera TCP

La longitud de cabecera TCP indica la cantidad de palabras de 32 bits contenidas en la cabecera. Esta información es necesaria porque el campo *Opciones* es de longitud variable, por lo que la cabecera también.

A continuación viene un campo de 6 bits que no se usan y, posteriormente seis banderas de un bit.

El campo *Tamaño de la ventana*, de 16 bits, informa al TCP receptor el número de bytes que el emisor está dispuesto a aceptar. El valor es este campo especifica el tamaño de la ventana deslizante.

También se proporciona una suma de comprobación para confiabilidad extrema. Es una suma de comprobación de cabecera, los datos y la pseudocabecera del TCP. Al realizar este cálculo, se establece el campo de suma de comprobación del TCP en cero, y se rellena el campo de datos con un byte cero adicional si la longitud es un número non. El algoritmo de suma de comprobación simplemente suma todas las palabras de 16 bits en complemento a 1 y

luego obtiene el complemento a 1 de la suma. Como consecuencia, al realizar el cálculo el receptor con el segmento completo, incluido el campo de suma de comprobación, el resultado debe ser 0.

La pseudocabecera mostrada en la figura 3.66, contiene las direcciones IP de 32 bits de las máquinas origen y de destino, el número de protocolo de TCP (6) y la cuenta de bytes del segmento. La inclusión de la pseudocabecera en el cálculo de la suma de comprobación TCP ayuda a detectar paquetes mal entregados.

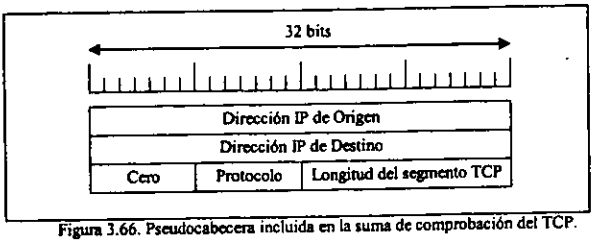


Figura 3.66. Pseudocabecera incluida en la suma de comprobación del TCP.

El campo *Apuntador Urgente*, de 16 bits, especifica una localización de byte en el área de datos TCP. El propósito de la bandera URG y del apuntador Urgente es informar al módulo TCP receptor que existe algún tipo de datos urgentes y señalar dónde se localizan.

El campo *opciones* se diseñó para contar con una manera de agregar características extras no cubiertas por la cabecera normal. La opción más importante es la que permite que cada anfitrión especifica la carga útil TCP máxima que pueda aceptar, es decir, el Tamaño Máximo de Segmento. Los módulos TCP sólo pueden utilizar esta opción en un mensaje que tenga establecida la bandera SYN. Uno de los lados de la conexión TCP anuncia al otro que espera un tamaño máximo de segmento de cierto valor. Si un módulo TCP no transmite un tamaño máximo de segmento, TCP por omisión, supone un tamaño de 536 bytes.

Establecimiento de una conexión TCP

Para establecer una conexión , el TCP utiliza un saludo (handshake) de tres etapas. El primer segmento de saludo se identifica porque en el campo de código tiene activo el bit SYN. El segundo mensaje tiene tanto el bit SYN como el bit ACK activos, indicando el acuse de recibido del primer segmento SYN y el hecho de que se continúa con el intercambio. El mensaje final del saludo es sólo un acuse de recibo y nada más utiliza para informar al destino que ambos extremos están de acuerdo en establecer una conexión.

El saludo de tres etapas es necesario para la sincronización correcta entre los datos extremos de la conexión. Vea la figura 3.57.

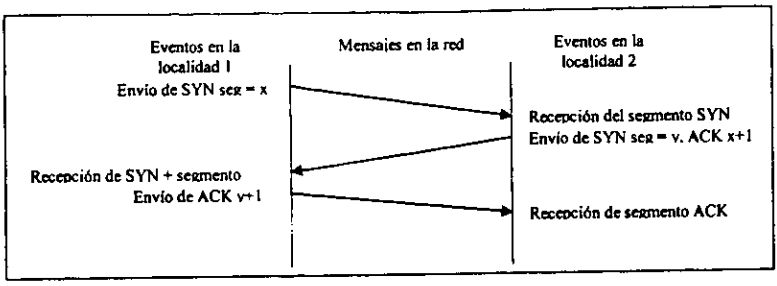


Figura 3.67. Saludo de tres etapas.

El saludo de tres etapas garantiza que ambos lados estén listos para transferir y permite a ambas partes, acordando un número de secuencia inicial. Los números de secuencia son enviados y reconocidos durante el saludo. Cada máquina debe seleccionar un número de secuencia inicial en forma aleatoria que se utilizará para identificar octetos en el flujo que esté enviando. Los números de secuencia no pueden comenzar siempre con el mismo valor, En particular, el TCP no puede seleccionar una secuencia 1 cada vez que crea una conexión.

El acuerdo que realizan las máquinas para establecer un número de secuencia para dos flujos después de tres mensajes solamente, se necesita de la información del campo de número de secuencia y del campo de acuse de recibo del segmento. La máquina A, que inicia un saludo, transfiere un número de secuencia inicial, x, en el campo de secuencia del primer segmento SYN como parte del saludo de tres etapas. La segunda máquina B, recibe el SYN, registra el número de secuencia y responde enviando su número de secuencia inicial en el campo de secuencia, así como un reconocimiento que especifica el byte x+1 esperado por B. En el mensaje final del saludo, A envía un acuse de recibo de la recepción del mensaje de B de todos los bytes a través de y. En todos los casos, los acuses de recibo siguen la convención de utilizar el número del próximo byte esperado.

Dos programas que utilizan el TCP para comunicarse pueden terminar la conversación valiéndose de la operación *close*. Cuando un programa de aplicación informa al TCP que ya no tiene más datos para enviar, éste cerrará 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 después, envía un segmento con el bit END activado. El receptor TCP reconoce el segmento END e informa el programa de aplicación en su extremo que no tiene más datos disponibles.

Una vez que la conexión se ha cerrado en una dirección dada, TCP rechaza más datos en esta dirección. Mientras tanto, los datos pueden continuar fluyendo en la dirección opuesta hasta que el emisor se cierra. Cuando ambas direcciones se han cerrado, el software TCP en cada punto extremo borra sus registros de la conexión.

3.6. Redes de Transmisión de Tramas Frame Relay.

La Recomendación I.122 de 1988 titulada Marco para Proporcionar Servicios Portadores en Modo Paquetes Adicionales, fue la presentación de una nueva forma de transmisión de paquetes por parte de los trabajos de RDSI[5], conocida como *Frame Relay* .

Frame Relay originalmente fue estandarizada por la UIT-T para optimizar el uso de los canales de RDSI en banda estrecha. En la actualidad, debido al desarrollo tecnológico que ha presentado Frame Relay, se ha convertido en una tecnología de red independiente de RDSI. Los planos de una arquitectura Frame Relay se presentan en la figura 3.70.

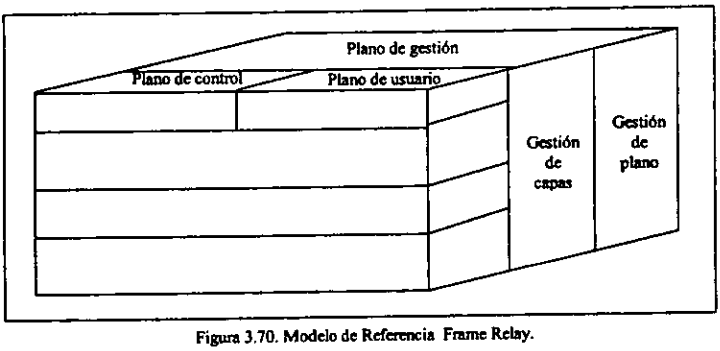


Figura 3.70. Modelo de Referencia Frame Relay.

En la actualidad, para la transmisión de información entre usuarios finales, el protocolo utilizado en el plano de usuario es el Q.922, una nueva recomendación, versión adaptada del protocolo LAP-D. Frame Relay sólo utiliza las funciones esenciales de este protocolo:

- Delimitación y transparencia de tramas
- Multiplexación y demultipexación de tramas utilizando el campo de dirección.
- Inspección de la trama para asegurar que está formada por un número entero de bytes antes de la inserción de un bit cero o después de la extracción de un bit cero.
- Inspección de la trama para comprobar que no es demasiado corta o demasiado larga.
- Detección de la transmisión de errores
- Funciones de control de congestión.

[5] Para mayor referencia vea apéndice A.

Todas las funciones anteriores se encontraban ya en el estándar LAP-D, anterior al Q.922 a excepción de la última, así como los campos de dirección de las tramas.

Estas funciones centrales de Q.922 en el plano de usuario constituyen un subnivel en el nivel de enlace. Proporcionan los servicios mínimos para la transmisión de las tramas de enlace desde un usuario a otro, sin tener en cuenta el control de flujo o el control de errores. Además de esto, el usuario puede elegir funciones adicionales extremo a extremo a nivel de enlace o de red, que no forman parte del servicio RDSI ofrecido. Basado en las funciones centrales, RDSI ofrece retransmisión de tramas como un servicio de nivel dos, orientado a conexión, con las siguientes propiedades:

- Preservación del orden de las tramas trasmitidas desde un extremo de la red a otro.
- Tramas no duplicadas.
- Pequeña probabilidad de pérdida de tramas.

En el plano de control, Q.922 proporciona un servicio de control de enlace de datos fiable, con control de errores y de flujo, a los mensajes de control de llamada I.451/Q.931 (que son también utilizados en RDSI). Esta arquitectura reduce al mínimo el trabajo a realizar por la red. Los datos de usuario se transmiten en tramas que prácticamente no son procesadas por los nodos intermedios en base al identificador de conexión. El proceso es como sigue: cuando una trama llega a un nodo, este automáticamente la envía a su destino una vez analizada la cabecera. Si ocurre un error, la transmisión se interrumpe. Si la trama está todavía en la red, los nodos se encargan de eliminarla; si hubiera llegado a su destino, el ETD mediante los protocolos de nivel superior, se encarga de solicitar la retransmisión. Frame Relay se concibió originalmente como un servicio opcional de RDSI. El usuario envía tramas al nodo de la red sobre un canal B, H o D y estas tramas se pasan al usuario de destino a través de la red. Sin embargo, las implementaciones reales de Frame Relay suelen ser independientes de RDSI.

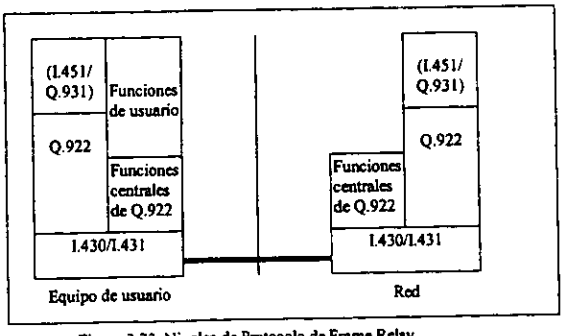


Figura 3.73. Niveles de Protocolo de Frame Relay

3.6.1. Comparación Frame Relay con X.25.

Como interfaz a una red, Frame Relay es parecido a X.25[6] en cuanto a su funcionalidad y formato. Sin embargo, Frame Relay es un protocolo más desarrollado, que proporciona un mejor servicio y una mayor eficiencia. En la figura 3.72 se presenta una comparación entre estos dos protocolos y el modelo OSI.

Los puntos principales en los que Frame Relay se diferencia de un servicio de conmutación de paquetes X.25 son:

- La señalización de control de llamada se realiza en una conexión lógica separada de la conexión para la transmisión de los datos de usuario.
- Como interfaz entre usuario y equipo de red, Frame Relay proporciona la multiplexación estadística para realizar conversaciones lógicas de datos (relacionados con circuitos virtuales) sobre un único enlace físico de transmisión.
- La multiplexación y conmutación de conexiones lógicas tiene lugar a nivel 2 en vez de a nivel 3, eliminando de esta manera un nivel entero de procesamiento.
- La red deja de preocuparse del control de errores y de control de flujo. Para emplear estos mecanismos, pasan a ser responsabilidad del nivel superior y se realizan extremo a extremo. Frame Relay incluye mecanismos muy simples de notificación de congestión, para permitir a una red informar a un dispositivo de usuario que los recursos de red están cerca de un estado de congestión. Esta notificación puede avisar a los protocolos de las capas más altas de que el control de flujo puede necesitarse.
- La velocidad de acceso que puede alcanzar una red Frame Relay es de 2 Mbps, frente a los 64 Kbps de X.25.

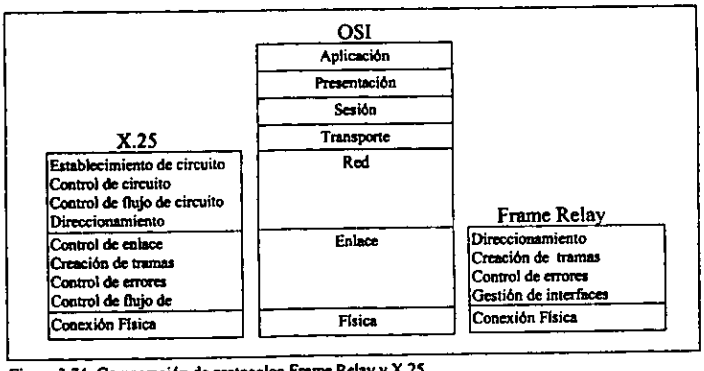


Figura 3.74. Comparación de protocolos Frame Relay y X.25.

[6] Para una mayor referencia ver en el apéndice A el tema del protocolo X.25.

3.6.2. Formato de Trama de Frame Relay.

El formato de la trama mostrado en la figura 3.75, es similar al del protocolo HDLC con la diferencia de que no hay un campo de control , por lo que:

- Sólo existe un tipo de trama, utilizada para transmitir información de usuario.
- No se puede utilizar señalización dentro de banda, una conexión lógica sólo puede transmitir datos de usuario.
- Tampoco existen tramas que permitan a la red ejecutar control de flujo, enviar acuses de recibido o perder retransmisiones, ya que no hay número de secuencia.
- La red detecta pero no recupera errores, los nodos de la red tienen capacidad de detectar errores y en determinados casos de eliminar tramas, pero nunca recuperarlos.

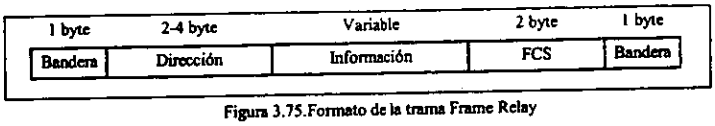


Figura 3.75.Formato de la trama Frame Relay

El *campo Bandera* contiene una secuencia de bits 01111110. Para garantizar la transparencia de la información, el nivel de enlace que va a transmitir la trama Frame Relay debe encargarse de comprobar el contenido de la trama entre el delimitador de apertura y de cierre e insertar un bit 0 cada vez que aparezca una secuencia de cinco bits 1 consecutivos. Por su parte el nivel de enlace de la entidad receptora se encargará de eliminar dichos bits una vez que obtenga los datos de la trama comprendidos entre ambos delimitadores.

El campo de dirección está formado por dos bytes pero puede extenderse hasta tres o cuatro. Los posibles formatos de este campo son:

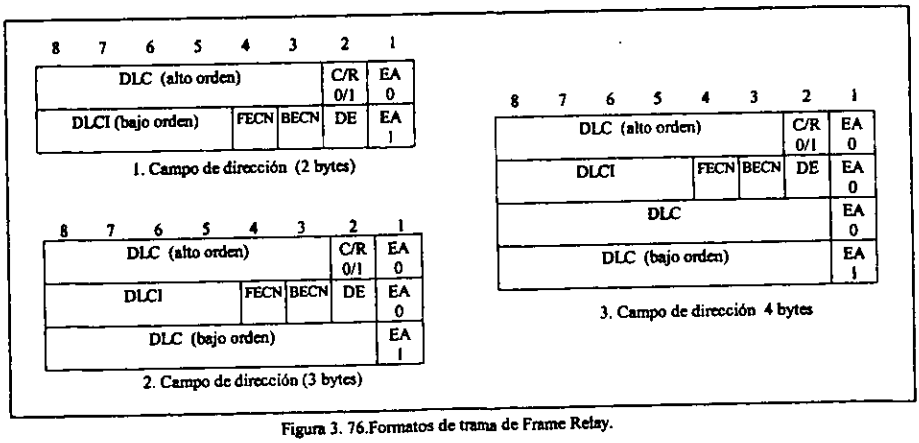


Figura 3. 76.Formatos de trama de Frame Relay.

Mantiene un identificador de conexión de enlace de datos DLCI de 10, 17 ó 24 bits, que permiten multiplexar conexiones lógicas Frame Relay sobre un único canal. El identificador de conexión tiene un significado puramente local; cada parte final de la conexión lógica asigna su propio DLCI, tomando un conjunto de números locales no utilizados, y la red se encarga de establecer su correspondencia.

Para Frame Relay en el canal D, se asume un campo de dirección de dos bytes y los valores de DLCI están limitados al rango 480 - 1,007. Esto equivale a un SAPI (identificadr de SAP, Punto de Acceso al Servicio) de 32 - 62.

La longitud del campo de dirección está definido por el campo *EA (Extended Address)* , que indica si el campo de dirección continúa en el siguiente byte (1) o ha terminado (0). El campo *C/R* es de uso específico en cada aplicación y el protocolo Frame Relay no lo utiliza.

El campo *Información* transmite datos del nivel superior. Si el usuario elige implementar funciones adicionales de control del nivel de enlace extremo a extremo, entonces en este campo encontraremos una trama de enlace de datos.

FCS (Frame -Check Sequence). Es una secuencia de 16 bits que permite verificar la correcta transmisión de la trama y la futura recuperación de posibles errores en la misma. Su funcionamiento es el mismo que en el protocolo HDLC.

La función de retransmisión de tramas realizadas por Frame Relay consiste en el ruteo de las tramas, antes descritas, de acuerdo a los valores de sus DLCI.

Por lo general, el ruteo es controlado mediante las entradas de una tabla de conexión que utiliza el DLCI. El manejador conmuta las tramas de un canal de entrada a otro de salida mediante la apropiada entrada de la tabla de conexión y traduce el DLCI de la trama antes de la transmisión.

Todas las terminales finales tienen una conexión lógica con valor DLCI = 0, que está reservada para el control de llamadas. Esto se utiliza cuando en el canal D no se usa LAPD para el control de llamadas.

Como parte de la función de retransmisión de tramas, se verifica el campo FCS de cada trama. Si se detecta un error, la trama simplemente se descarta, siendo responsabilidad de los usuarios finales la recuperación de este error.

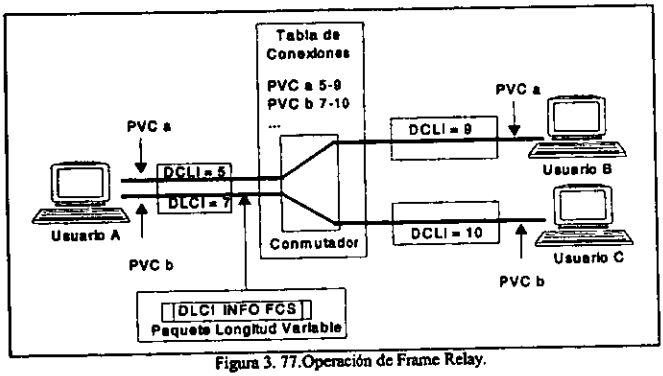


Figura 3. 77.Operación de Frame Relay.

En la figura 3.77, el usuario A desea comunicarse con el usuario B. Primero deberá asegurarse que dispone de un circuito virtual (CV) que una a ambas usuarios. La información, antes de ser entregada a la red, deberá ser

segmentada en tramas a las que le añade un identificador común llamado DLCI. Ya en la red las tramas son conmutadas de acuerdo con unas tablas de ruteo que asocian cada DLCI de entrada con un puerto de salida y un nuevo DLCI, hasta que llegan a su destino donde son de nuevo ensambladas.

Control de Gestión

Una red Frame Relay es una red de conmutación de paquetes en la que los paquetes son tramas de nivel 2. Básicamente Frame Relay es una red de colas. En cada manejador, hay una cola de tramas por cada enlace de la salida. Si la velocidad de llegada de las tramas excede la velocidad de transmisión de las mismas, el tamaño de la cola crece sin límite y el retraso sufrido por una trama tiende a infinito. Por otra parte, si la velocidad de llegada de la tramas es menor que la velocidad de transmisión, la longitud de la cola crecerá muy rápidamente a medida que la velocidad de llegada se aproxime a la velocidad de retransmisión.

Cualquier manejador tiene conectado un determinado número de enlaces de transmisión a otros manejadores y directamente a usuarios finales. En cada enlace, las tramas entran y salen. Puede considerarse que hay dos buffers de tamaño variable en cada enlace: uno que recibe las tramas que llegan y otro que guarda las tramas que están esperando ser transmitidas, con la única limitación de que la suma de sus tamaños debe ser siempre constante. Vea la figura 3.78.

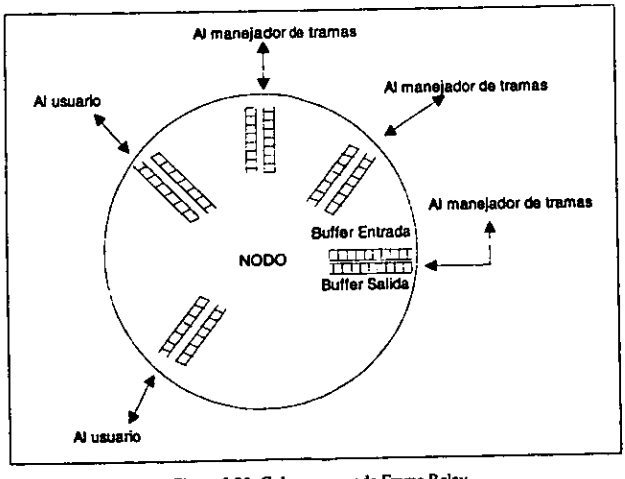


Figura 3.78. Colas en un nodo Frame Relay

De cualquier manera, cuando llega una trama, se almacena en el buffer de entrada del enlace correspondiente. El manejador examina cada trama de entrada para tomar una decisión de ruteo y entonces mueve dicha trama al buffer de salida más apropiado. Las tramas encoladas para salir se transmiten tan rápidamente como sea posible. Pero si las tramas llegan demasiado rápido al manejador para que éste pueda procesarlas, o llegan más rápido de lo que parten

las tramas de los buffers de salida, entonces habrá un momento en el que no se dispondrá de memoria para las nuevas tramas de entrada.

Cuando se alcanza este punto de saturación, se pueden adoptar dos estrategias. La primera consiste en dascartar cualquier trama de entrada para la que no haya espacio en el buffer. Pero este método no es aconsejable, ya que las tramas descartadas deben ser retransmitidas, aumentando de este modo la congestión de la red. La otra alternativa es utilizar algún mecanismo que limite la velocidad a la que las nuevas tramas entran en la red. Este procedimiento es conocido como control de congestión.

UIT-T en la serie I3xx, define los objetivos del control de congestión en Frame Relay de la siguiente manera:

- Minimizar el descarte de tramas.
- Mantener, con una probabilidad alta y mínima variación, la calidad de servicio acordada.
- Minimizar la posibilidad de que un usuario monopolice los recursos de la red a expensas de otros usuarios.
- Facilidad de implementación y suponer poca carga para los usuarios finales de la red.
- Crear el menor tráfico adicional posible en la red.
- Distribuir los recursos de la red equitativamente entre los usuarios.
- Limitar la transmisión de la congestión a otras redes y elementos dentro de la red.
- Operar con efectividad, sin depender del flujo del tráfico, en cualquier dirección entre los usuarios finales.
- Tener la mínima interacción, o con impacto sobre, otros sistemas en la red Frame Relay.
- Minimizar la variación de la calidad del servicio debida a las conexiones Frame Relay individuales durante la congestión.

El control de congestión es una responsabilidad compartida entre la red y los usuarios finales. La red es la que mejor puede monitorizar el grado de congestión, mientras que los usuarios son los que mejor pueden controlar esta congestión limitado el tráfico. Teniendo esto en cuenta, se consideraran dos estrategias generales para el control de congestión:

Los procedimientos para evitar la congestión se utilizan cuando ésta se inicia, a fin de minimizar sus efectos sobre la red. Estos procedimientos son mecanismos de señalización explícita.

Los procedimientos de recuperación de la congestión se utilizan para prevenir el colapso de la red en la fase de congestión severa. Se inicia generalmente cuando la red empieza a eliminar tramas debido a la congestión. Estas tramas sirven como mecanismo de señalización implícita.

Procedimientos con señalización explícita

En el campo de direccionamiento se dispone de dos bits de señalización explícita. Cuando un manejador de tramas detecta congestión en la red, hace uso de los dos bits de señalización explícita del campo de direccionamiento para

indicarles a los usuarios finales que deben realizar los procedimiento necesarios para evitar dicha congestión. Estos bits son:

- *BECN (Backward explicit congestion notification).* Indica al usuario que deben iniciarse procedimientos para evitar la congestión en la dirección opuesta a la de la trama recibida.
- *FECN (Forward explicit congestion notification).* Indica al usuario que deben iniciarse procedimientos para evitar la congestión del tráfico en la misma dirección que la trama recibida. Indica que esta trama, en conexión lógica, ha encontrado recursos que sufren congestión.

En primer lugar, en el caso de la red, es necesario que cada menejador de tramas controle la ocupación de sus colas. Si la longitud de una cola empieza a alcanzar un nivel peligroso, se indica en el bit BECN, en el bit FECN o una combinación de ambos, para intentar reducir el flujo de tramas que atraviesa el manejador. La elección del bit BECN o FECN depende de si los usuarios finales de una determinada conexión lógica están preparados para responder a uno u otro de estos bits. Esto se determina en la fase de configuración. En cualquier caso, el manejador de tramas elige qué conexión lógica debe ser prevenida del peligro de congestión. En las primeras etapas de la congestión, sólo realiza la notificación a los usuarios de aquellas conexiones que están generando el mayor tráfico, mientras que si la congestión comienza a ser importante, la notificación se produce en todas las conexiones.
La respuesta del usuario depende de la llegada de las señales BECN o FECN. El procedimiento más simple es el de respuesta a una señal BECN, en este caso el usuario simplemente reduce la velocidad de transmisión de tramas hasta que la señal lo indica. La respuesta a un FECN es más complicada, ya que requiere que el usuario local pida al usuario del otro extremo de la conexión que reduzca su flujo de tramas. Las funciones principales utilizadas en el protocolo Frame Relay no soportan este tipo de notificación, por lo que se deja esta tarea a los niveles superiores. El control de flujo puede llevarse a cabo mediante Q.922 ó cualquier otro protocolo de control de enlace implementado sobre el subnivel de Frame Relay.

Procedimientos de recuperación de la congestión con señalización implícita

La señalización implícita se produce cuando la red descarta una trama y el usuario final a un nivel superior detecta este hecho. Cuando esto ocurre, los niveles superiores del usuario final pueden deducir que existe congestión. Por ejemplo, en un protocolo de control de enlace de datos como Q.922, cuando se descarta una trama por una sobrecarga de buffer en la red, la siguiente trama generará una trama Reject desde el punto final receptor. Por lo tanto, se puede utilizar un procedimiento de nivel superior que proporcione control de flujo para recuperarse de la congestión.
El papel de la red es el de descartar tramas según sea necesario y para ello se puede utilizar un bit que existe en el campo de dirección de todas las tramas, *DE (Discard eligibility).* Cuando sea necesario descartar tramas, aquellas que tengan este bit a 1 tienen preferencia sobre las que tienen este bit a 0.

Esta capacidad de utilización del bit *DE* hace posible que el usuario temporalmente puede mandar más tramas de las que tiene permitidas en promedio. En este caso, el usuario fija el bit *DE* en las tramas que exceden el promedio y la red las retransmite si es posible.

Los manejadores de tramas también pueden fijar este bit, si deciden que la entrada de tramas que provienen del usuario es potencialmente excesiva.

El mecanismo funciona de la siguiente manera: cada usuario puede negociar una velocidad de información comprometida, *CIR (commited information rate)* en bps, durante el tiempo de configuración de la conexión. Este *CIR* requerido representa la estimación de lo que será su tráfico normal durante un período de ocupación. El *CIR* concedido, que es menor o igual al CIR requerido, es el compromiso de la red para transmitir datos a esa velocidad si no se producen errores. El manejador de tramas al que está conectada la estación de usuario realiza una función de filtro.

Si el usuario envía datos a menor velocidad que la que marca el *CIR*, el manejador de tramas de entrada no varía el valor del bit DE. Si la velocidad es superior al *CIR*, el manejador marca los bits *DE* de las tramas que superan el promedio y los envía a la red; estas tramas pueden alcanzar su destino o ser descartadas si se produce congestión. Por último, se define una velocidad máxima, por encima de la cual todas las tramas se descartan en su entrada al manejador.

El utilizar Frame Relay como protocolo interno de la red proporciona las ventajas señaladas de que la red descarta las tramas erróneas y no realiza funciones de control de error ni control de flujo. En consecuencia, son necesarias pocas instrucciones para enviar y recibir datos sobre los enlaces intermedios. Así mismo, los buffers requeridos en los nodos intermedios se reducen considerablemente, pues no es necesario almacenar las tramas hasta que se reciba un reconocimiento positivo.

3.6.5. Implementación en Red.

Frame Relay puede ser usada como una interfaz a un proveedor público de servicios o a una red de equipo privado. Un método de implementación de red privada es equipar los multiplexores T1 con interfaces Frame Relay para dispositivos de datos. Estos dispositivos que se observan en la figura 3.79, son conocidos como ruteadores FR ó FRAD (Frame Relay Access Devices, dispositivos de Acceso Frame Relay).

Las redes Frame Relay suelen proporcionar la conversión X.25 - Frame Relay, con lo que facilitan la migración y coexistencia. Otros servicios que no requieren de una interfaz Frame Relay son: voz, video y teleconferencia.

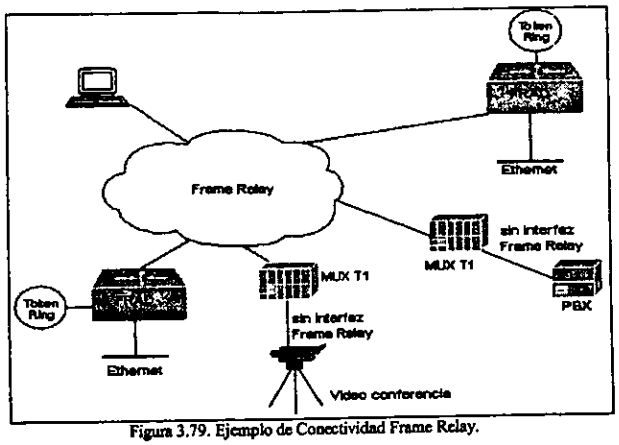


Figura 3.79. Ejemplo de Conectividad Frame Relay.

Un servicio público Frame Relay es desplegado poniendo equipos Frame Relay de conmutación en las oficinas centrales de un proveedor de telecomunicaciones. En este caso, los usuarios pueden obtener beneficios económicos de las limitaciones de cargos sensibles al tráfico, y son relevados del trabajo necesario que conlleva administrar y mantener el equipo de red y el servicio.

3.7. IPv6.

La actual red Internet, basada en un diseño de principios de los años 80, ha experimentado un gran crecimiento tanto en el número de usuarios conectados como en aplicaciones y servicios disponibles, sin embargo han aparecido deficiencias en los aspectos administrativos y de seguridad, así como carencias en la prestación de los servicios avanzados que se requieren naturalmente.

Para remediar estas carencias, los cuerpos técnicos de la Internet impulsaron un debate bajo el lema de IP Next Generation (IPng) que ha culminado con la especificación de un nuevo protocolo IP, sucesor del actual IPv4, conocido formalmente como la versión 6 del Protocolo Internet o IPv6.

El 1990, el IETF comenzó a trabajar sobre una nueva versión del IP, una que nunca se quedaría sin direcciones, resolvería varios otros problemas y sería más flexible y eficiente también. Sus metas principales eran:

1. Manejar miles de millones de hosts, aun con asignación de espacio de direcciones ineficiente.
2. Reducir el tamaño de las tablas de ruteo.
3. Simplificar el protocolo, para permitir a los rutadores el procesamiento más rápido de los paquetes.
4. Proporcionar mayor seguridad (verificación de autenticidad y confidencialidad) que el IP actual.
5. Prestar mayor atención al tipo de servicio, especialmente con datos en tiempo real.

6. Ayudar a la multitransmisión permitiendo la especificación de alcances.
7. Posibilitar que un host sea móvil sin cambiar su dirección.
8. Permitir que el protocolo evolucione.
9. Permitir que el protocolo viejo y el nuevo coexistan por años.

Para encontrar un protocolo que cumpliera con todos estos requisitos, la IETF hizo una convocatoria solicitando propuestas y estudios en el RFC 1550. Se recibieron 21 respuestas, no propuestas completas. En diciembre de 1992 había siete propuestas serias en la mesa; iban desde hacer cambios menores al IP hasta desecharlo y reemplazarlo por un protocolo completamente diferente.

Tres de las mejores propuestas se publicaron en IEEE Network (Deering, 1993; Francis, Katz y Ford, 1993). Tras muchos análisis y revisiones, se seleccionó una versión modificada de la combinación de las propuestas de Deering y Francis, llamada ahora SIPP (Simple Internet Protocol Plus, protocolo simple de Internet mejorado), y se le dio la designación IPv6 (el IPv5 ya estaba en uso con un protocolo experimental de flujos en tiempo real)

En general, IPv6 no es compatible con IPv4, pero es compatible con todos los demás protocolos Internet, incluidos TCP, UDP, IGMP, OSPF, BGP y DNS, a veces requiriendo pequeñas modificaciones (principalmente para manejar direcciones más grandes). Las características principales del IPv6 se analizan a continuación. Puede encontrarse mayor información en los RFC 1883 a 1887.

Por principio, y lo más importante, el IPv6 tiene direcciones más grandes que el IPv4; son de 16 bytes de longitud, lo que resuelve el problema del actual límite en el número de direcciones.

La segunda mejora principal del IPv6 es la simplificación de la cabecera, que contiene sólo dos (contra 13 en el IPv4). Este cambio permite a los ruteadores procesar con mayor rapidez los paquetes y mejorar, por tanto, el rendimiento.

La tercera mejora importante fue un mejor apoyo de opciones. Este cambio fue esencial en la nueva cabecera, pues campos que antes eran obligatorios ahora son opcionales. Además, es diferente la manera de representar las opciones, haciendo más sencillo que los ruteadores hagan caso omiso de opciones no dirigidas a ellos. Esta característica mejora el tiempo de procesamiento de los paquetes.

Una cuarta área en la que el IPv6 representa un avance importante es la seguridad. La IETF tenía infinidad de historias sobre adolescentes que usaban sus computadoras personales para meterse en bancos e instalaciones militares por todas partes de Internet. Se tenía la sensación de que había que hacerse algo para mejorar la seguridad. Las verificaciones de autenticidad y la confidencialidad son características clave del IP nuevo.

Por ultimo, se proporciona una mayor atención al tipo de servicio que en el pasado. El IPv4 de hecho tiene un campo de 8 bits dedicado a este asunto, pero para el crecimiento esperado del tráfico multimedia en el futuro, se requiere mucho más.

La cabecera principal del IP6

El IPv6 cambia completamente el formato de datagrama. Un datagrama IPv6 tiene un encabezado base de tamaño fijo, seguido por ceros o más encabezados de extensión, seguidos a su vez por datos. La cabecera del IPv6 se muestra en la siguiente figura.

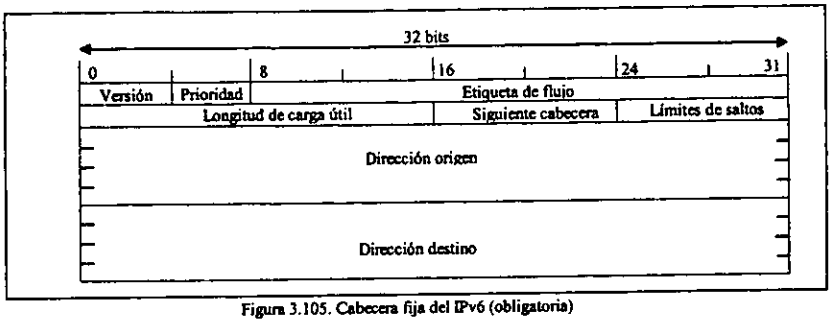


Figura 3.105. Cabecera fija del IPv6 (obligatoria)

El campo *versión* ocupa 4 bits, y contiene el número de versión del IP, en este caso 6. Tal vez durante el periodo de transición del IPv4 al IPv6, los ruteadores podrán examinar este campo para saber el tipo de paquete que tienen.

El campo de *prioridad* ocupa 4 bits e indica la importancia del paquete que se esta enviando. Los valores 0 a 7 son para transmisiones capaces de reducir su velocidad en caso de un congestionamiento. Los valores 8 a 15 son para tráfico de tiempo real cuya tasa de envío es constante, aun si todos los paquetes se están perdiendo. El audio y el video caen en esta ultima categoría. Esta distinción permite a los ruteadores manejar mejor los paquetes en caso de congestionamientos. Dentro de cada grupo, los paquetes de número más bajo son menos importantes que los paquetes de número alto. El estándar IPv6 sugiere, por ejemplo, usar 1 para noticias, 4 para FTP y 6 para conexiones Telnet, pues el retardo de un paquete de noticias durante algunos segundos no es perceptible, pero el retardo de un paquete Telnet ciertamente lo es.

El uso del campo *etiqueta de flujo* será para indicar que el paquete requiere un tratamiento especial por parte de los ruteadores que lo soporten. Cuando aparece un paquete con una etiqueta de flujo diferente de cero, todos los ruteadores pueden buscar en sus tablas internas para ver el tipo de tratamiento especial que requiere. En efecto, los flujos son un intento de tener lo mejor de ambos mundos: la flexibilidad de una subred de datagramas y las garantías de una subred de circuitos virtuales.

El campo de *longitud de carga útil* ocupa 16 bits e indica la longitud en bytes de los datos del mensaje. El campo *siguiente cabecera* de 8 bits indica a que protocolo corresponde la cabecera que está a continuación de la actual. Si esta cabecera es la última cabecera de IP, el campo de *siguiente cabecera* indica el manejador de protocolo de transporte (por ejemplo TCP, UDP) al que se entregara el paquete.

El campo *límite de saltos* se usa para evitar que los paquetes vivan eternamente. En practica es igual al campo de tiempo de vida del IPv4, es decir, un campo que se disminuye cada salto. En teoría, en el IPv4 era un tiempo en segundos, pero ningún ruteador lo usaba de esa manera, por lo que se cambio el nombre para reflejar la manera en que se usa realmente.

Luego vienen los campos de dirección de origen y dirección de destino. La propuesta original de Deering, el SIP, usaba direcciones de 8 bytes, pero durante el periodo de revisión muchas personas sintieron que, con direcciones de 8 bytes, en algunas décadas el IPv6 quedaría sin direcciones, y que con direcciones de 16 bytes nunca se acabarían. Otros argumentaban que 16 bytes era demasiado, mientras que unos más estaban a favor de usar dirección de 20 bytes para hacerlas compatibles con el protocolo de datagramas de OSI. Otra fracción quería direcciones de tamaño variable. Tras mucho debate, se decidió que la mejor medida sería una dirección de 16 bytes de longitud fija.

Además de manejar direcciones unitransmisión estándar (punto a punto) y multitransmisión, el IPv6 también maneja un nuevo tipo de direccionamiento: transmisión a cualquiera. La transmisión a cualquiera (anycasting) es como la multitransmisión, en el sentido de que el destino es un grupo de direcciones, pero en lugar de tratar de entregar el paquete a todos, intenta entregarlo a uno solo, generalmente al más cercano. Por ejemplo, al ponerse en contacto con un grupo de servidores de archivos cooperativos, un cliente puede usar la transmisión a cualquiera para llegar al más cercano, sin tener que saber quien es. La transmisión a cualquiera usa direcciones normales de unitransmisión. Es responsabilidad del sistema de ruteado escoger al host que recibirá el paquete.

En IPv6 se ha desarrollado una nueva notación para escribir direcciones de 16 bytes: se escriben como ocho grupos de cuatro dígitos hexadecimales, separados los grupos por dos puntos, como sigue:

8000:0000:0000:0000:0123:4567:89AB:CDEF

Ya que muchas direcciones tendrán muchos ceros en ellas, se han autorizado tres optimizaciones. Primero, los ceros a la izquierda de un grupo pueden omitirse, por lo que 0123 puede escribirse como 123. Segundo, pueden reemplazarse uno o más grupos de 16 ceros por un par de signos de dos puntos. Por tanto, la dirección anterior se vuelve ahora

8000::0123:4567:89AB:CDEF

Por último, las direcciones IPv4 pueden escribirse como un par de signos de dos puntos y un número decimal anterior separado por puntos, como por ejemplo:

:: 192.31.20.46

Cabeceras de extensión

Estas cabeceras pueden usarse para proporcionar información extra, pero codificada de una manera eficiente. Hay seis tipos de cabeceras de extensión definidas actualmente, que se listan en la tabla 3.22. Todas son opcionales, pero si hay más de una, deben aparecer justo después de la cabecera fija, y de preferencia en el orden listado.

Cabecera de extensión	Descripción
Opciones salto por salto	Información diversa para los ruteadores
Ruteo	Ruta total o parcial a seguir
Fragmentación	Manejo de fragmentos de datagramas
Verificación de autenticidad	Comprobación de la identidad del transmisor
Carga útil de seguridad	Información sobre el contenido cifrado
Opciones de destino	Información adicional para el destino

Tabla 3.22. Cabeceras de extensión del IPv6

Algunas de las cabeceras tienen un formato fijo; otras contienen un número variable de campos de longitud variable. En estos, cada elemento esta codificado como una tupla (tipo, longitud, valor). El tipo es un campo de 1 byte que indica la opción de la que se trata. Los valores de tipo se han escogido de modo que los dos primeros bits indican a los ruteadores que no saben como procesar la opción lo que tienen que hacer. Las posibilidades son: saltar la opción, descartar el paquete, descartar el paquete y enviar de regreso un paquete ICMP, y lo mismo que lo anterior, pero no enviar paquetes ICMP a direcciones de multitransmisión (para evitar que un paquete de multitransmisión malo genere millones de informes ICMP).

- La *longitud* también es un campo de 1 byte, e indica la longitud del valor (0 a 255 bytes). El valor es cualquier información requerida, de hasta 255 bytes.

La cabecera de salto por salto se usa para informar que se deben examinar todos los ruteadores a lo largo de la trayectoria. Hasta ahora, se ha definido una opción: manejo de datagramas de más de 64K. El formato de esta cabecera se muestra en la figura 3.106.

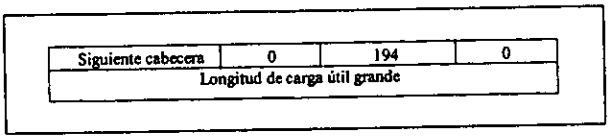


Figura 3.106. La cabecera de extensión salto por salto para datagramas grandes (jumbogramas).

Como con todas las cabeceras de extensión, esta comienza con 1 byte que indica el tipo de cabecera que sigue. A este byte le sigue uno que indica la longitud de la cabecera salto por salto en bytes, excluyendo los primeros 8 bytes, que son obligatorios. Los 2 bytes siguientes indican que ésta opción define el tamaño del datagrama (código 194) como número de 4 bytes. Los últimos 4 bytes indican el tamaño del datagrama. No se permiten los tamaños menores que 65,536, que darán como resultado el que el primer ruteador descarte el paquete y devuelva un mensaje ICMP de error. Los datagramas que usan esta cabecera de extensión se llaman ***jumbogramas***. El uso de jumbogramas es importante para las aplicaciones de supercomputadoras que deben transferir con eficiencia gigabytes de datos a través de Internet.

La cabecera de ruteo lista uno o más ruteadores que deben visitarse en el camino al destino. Hay la posibilidad tanto de ruteo estricto (se suministra la trayectoria completa) como libre (solo se proporcionan ruteadores seleccionados), pero están combinados. El formato de la cabecera de ruteo se muestra en la figura 3.107.

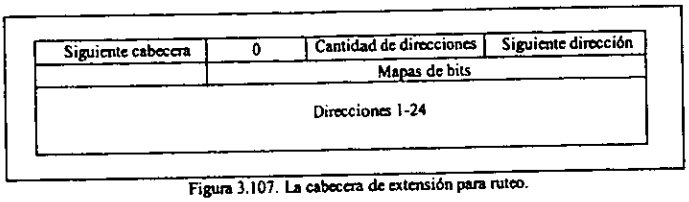


Figura 3.107. La cabecera de extensión para ruteo.

Los primeros 4 bytes de la cabecera de extensión de ruteo contienen cuatro enteros de 1 byte: el tipo de la siguiente cabecera, el tipo de ruteo (actualmente 0), la cantidad de direcciones presentes en esta cabecera (1 a 24) y el índice de la siguiente dirección a visitar. Este último campo comienza en 0 y aumenta con cada dirección que se visita.

Después viene un byte reservado seguido de un mapa de bits con bits para cada una de las 24 direcciones IPv6 potenciales que siguen. Estos bits indican si debe visitarse cada dirección directamente después de la anterior (ruteo estricto desde el origen) o si pueden intervenir otros ruteadores intermedios (ruteo libre desde el origen).

La cabecera de fragmentación maneja la fragmentación de una manera parecida a la del IPv4. La cabecera contiene el identificador del datagrama, el número de fragmento y un bit que indica si seguirán más fragmentos. En el IPv6, a diferencia del IPv4, sólo el host de origen puede fragmentar un paquete. Los ruteadores a lo largo del camino no pueden hacerlo.

La cabecera de verificación de autenticidad proporciona un mecanismo mediante el cual el receptor de un paquete puede estar seguro de quien lo envió. Con el IPv4 no existe tal garantía. La carga cifrada de seguridad posibilita el cifrado del contenido de un paquete de modo que sólo el receptor pretendido pueda leerlo. Estas cabeceras usan técnicas de cifrado para lograr su cometido.

Cuando un transmisor y un receptor desean comunicarse con seguridad, primero deben ponerse de acuerdo en una o más claves secretas que solo ellos conocen. La manera en que lo hacen no es asunto del IPv6. A cada una de estas claves se le asigna un número de clave único de 32 bits. Los números de clave son globales, por lo que si un usuario A está usando la clave 4 para hablar con un usuario B, no puede usar una clave 4 para hablar con un usuario C. Asociados a cada número de clave hay otros parámetros, como tiempo de vida de la clave y otros.

Para enviar un mensaje con verificación de autenticidad, el transmisor primero construye un paquete que consiste en todas las cabeceras IP y la carga útil, y luego reemplaza los campos que cambian en el camino (por ejemplo, el limite de saltos) por ceros. Se rellena el paquete con ceros hasta un múltiplo de 16 bytes. Del mismo modo, la clave secreta a usar se rellena con ceros hasta un múltiplo de 16 bytes. Ahora se calcula una suma de comprobación cifrada con base en la concatenación de la clave secreta rellenada, el paquete rellenado y nuevamente la clave secreta rellenada.

Los usuarios pueden definir su propio algoritmo de suma de comprobación cifrada, pero se recomienda a los usuarios con poca habilidad en el cifrado usar el algoritmo predeterminado, el MD5.

Por otra parte, la cabecera de verificación de autenticidad básicamente está contenida por tres partes. La primera consiste en 4 bytes que contienen el siguiente número de cabecera, la longitud de la cabecera de verificación de autenticidad, y 16 bits cero. Después viene el número de clave de 32 bits. Por ultimo, se incluye la suma de comprobación MD5.

El receptor entonces usa el número de clave para encontrar la clave secreta. La versión rellenada de ella se agrega al principio y al final de la carga útil rellenada, los campos de cabecera variables se llenan con ceros, y se calcula la suma de comprobación; si esta de acuerdo con la suma incluida en la cabecera de verificación de autenticidad, el receptor puede estar seguro de que el paquete llego del transmisor con quien comparte la clave secreta, y también puede estar seguro de que el paquete no fue alterado en el camino. Las propiedades del MD5 hacen imposible, desde el punto de vista de cómputo, que un intruso falsifique la identidad del transmisor o modifique el paquete de alguna manera que pueda escapar a la detección.

Es importante apuntar que la carga útil de un paquete con verificación de autenticidad se envía sin cifrado. Cualquier ruteador a lo largo del camino puede leer lo que dice. En muchas aplicaciones el secreto no es realmente importante, solo la relación de autenticidad. Por ejemplo, si un usuario indica a su banco que pague su cuenta telefónica, probablemente no habrá necesidad real del secreto, pero hay una necesidad muy real de que el banco tenga la seguridad completa de quien envió el paquete que contiene la orden de pago.

Para paquetes que deben enviarse secretamente, se usa la cabecera de extensión de carga útil cifrada de seguridad. Esta comienza con un número de clave de 32 bits, seguido de la carga cifrada. El algoritmo de cifrado es responsabilidad del transmisor y del receptor, pero por omisión es DES en modo de encadenamiento de bloque cifrado (CBC). Al usarse DES-CBC, el campo de carga útil comienza por el vector de inicialización (un múltiplo de 4 bytes), luego la carga útil, y luego el relleno a un múltiplo de 8 bytes. Si se desean tanto cifrado como verificación de autenticidad, se requieren ambas cabeceras.

Una controversia relacionada con la seguridad tiene que ver con la selección de los algoritmos predeterminados que todas las implementaciones deben reconocer. Si bien que se pensaba que el MD5 era relativamente seguro, los avances recientes en cifrado pueden debilitarlo. Ningún criptografo serio cree que el DES es seguro ante ataques de grandes gobiernos, pero probablemente es lo bastante bueno como para frustrar incluso al adolescente que lo intenten, cuando menos por ahora. El arreglo fue entonces integrar seguridad en el IPv6, usar un algoritmo de suma de comprobación de primer nivel para una buena verificación de autenticidad y un algoritmo no tan robusto para el secreto, pero dar al usuario la opción de reemplazar estos algoritmos por los suyos propios.

La intención de la cabecera de opciones de destine es incluir campos que sólo deben interpretarse en el host de destino. En la versión inicial de IPv6, las únicas opciones definidas son las opciones nulas para rellenar esta cabecera a un múltiplo de 8 bytes, por lo que no se usara inicialmente; se incluyo para asegurar que el nuevo software de y de los hosts puedan manejarla, en caso de que a alguien se le ocurra una opción de destino algún día.

Capítulo 4

Seguridad en Redes de Computadoras

Las redes de computadoras se han convertido en un medio muy importante en la comunicación de datos. Actualmente, se ha incrementado la variedad y cantidad de usuarios que usan la red para diversos fines (el aprendizaje, la docencia, la investigación, la búsqueda de socios o mercados, el esparcimiento, entre otros), así también, han surgido redes de alta velocidad con el objetivo de hacer más eficiente la transmisión de información.

A pesar de los beneficios que brinda este tipo de tecnología su uso ofrece riesgos en contra de la integridad y confiabilidad de la información que se maneja, afectando los sistemas y recursos de los usuarios.

4.1. Seguridad.

La Organización Internacional de Estandarización (ISO - International Organization for Standarization) publicó a principios de los ochenta la normativa ISO 7498 en la que se definía un modelo estándar de interconexión de sistemas abiertos (OSI-Open Systems Interconnection) que permite la comunicación entre diferentes sistemas con diferentes equipos y sistemas operativos.
En 1988 se publicó una segunda parte de la normativa OSI, el documento OSI 7498-2, donde se definen los servicios de seguridad de la arquitectura (Security Architecture), para proteger las comunicaciones de los usuarios en las redes. Estos servicios de seguridad son los siguientes:

1. **Autentificación de entidad par.** Servicio que confirma la fuente de una unidad de datos. La autentificación puede ser sólo de la entidad origen o de la entidad destino, o ambas entidades se pueden autentificar la una a la otra.
2. **Control de acceso.** Servicio que se utiliza para evitar el uso no autorizado de recursos.
3. **Confidencialidad de datos.** Este servicio proporciona protección contra la revelación deliberada o accidental de los datos en una comunicación.
4. **Integridad de datos.** Este servicio garantiza que los datos recibidos por el receptor de una comunicación coinciden con los enviados por el emisor.
5. **No repudio.** Este servicio proporciona la prueba ante una tercera parte de que cada una de las entidades comunicantes han participado en la comunicación y puede ser de dos tipos:

    - *Con prueba de origen.* Cuando el destinatario tiene prueba del origen de los datos.
    - *Con prueba de entrega.* Cuando el origen tiene prueba de la entrega íntegra de los datos al destinatario deseado.

Para proporcionar estos servicios de seguridad es necesario incorporar en los niveles apropiados del Modelo de Referencia OSI mecanismos de seguridad como:

1. **Cifrado.** El cifrado consiste en convertir un mensaje comprensible en uno secreto, utilizando sistemas criptográficos simétricos o asimétricos y se puede aplicar extremo a extremo o individualmente a cada enlace del sistema de comunicaciones.
2. **Firma digital.** Se define como el conjunto de datos que se añaden a una unidad de datos para protegerlos contra la falsificación, permitiendo al receptor probar la fuente y la integridad de los mismos. La firma digital supone el cifrado, con una componente secreta del firmante, de la unidad de datos y la elaboración de un valor de control criptográfico.
3. **Control de acceso.** Este mecanismo se utiliza para autentificar las capacidades de una entidad, con el fin de asegurar los derechos de acceso a recursos que posee. El control de acceso se puede realizar en el origen o en un

punto intermedio, y se encarga de asegurar si el emisor está autorizado a comunicarse con el receptor y a usar los recursos de comunicación requeridos. Si una entidad intenta acceder a un recurso no autorizado, o intenta el acceso de forma impropia a un recurso autorizado, entonces la función de control de acceso rechazará el intento y registrará el incidente con el propósito de generar una alarma.

4. **Integridad de datos**. Para proporcionar la integridad de una unidad de datos la entidad emisora añade a la unidad de datos una cantidad que se calcula en función de los datos. Esta cantidad, probablemente encriptada con técnicas simétricas o asimétricas, puede ser una información suplementaria compuesta por un código de control de bloque, o un valor de control criptográfico. La entidad receptora genera la misma cantidad a partir del texto original y la compara con la recibida para determinar si los datos no se han modificado durante la transmisión. Por otra parte, para proporcionar integridad a una secuencia de unidades de datos se requiere, adicionalmente, alguna forma de ordenación explícita, tal como la numeración de secuencia, un sello de tiempo o un encadenamiento criptográfico.
5. **Intercambio de autentificación**. Existen dos mecanismos de autentificación:

    - *Autentificación simple*. El emisor envía su nombre distintivo y una contraseña al receptor, el cual los comprueba.
    - *Autentificación fuerte*. Utiliza las propiedades de los criptosistemas de llave pública. Cada usuario se identifica por un nombre distintivo y por su llave secreta. Cuando un segundo usuario desea comprobar la autenticidad de su interlocutor deberá comprobar que éste está en posesión de su llave secreta, para lo cual deberá obtener su llave pública.

Básicamente, dichos servicios tratan aspectos como la confidencialidad, la integridad y el control de accesos y se señala al cifrado como uno de los mecanismos más adecuado para conseguir los objetivos de seguridad que se pretenden alcanzar.

En los siguientes temas se mostrarán algunos algoritmos, protocolos y productos comerciales que implementan los servicios de seguridad que se definen en OSI 7498-2.

4.2. Criptografía.

Una de las primeras escrituras secretas de que se tiene conocimiento data del siglo V a. C., durante la guerra entre Atenas y Esparta. Este método de cifrado consistía en la alteración del mensaje mediante la inclusión de símbolos innecesarios que desaparecían al enrollar la lista en un rodillo, de manera que se mostraban sólo los símbolos del mensaje.

Posteriormente, empezó la creación de diferentes cifrados con el propósito de que la información de interés llegara a su destino de manera oculta. Uno de los cifrado más conocidos y sencillos de implementar es el método de César, consistía en una sustitución de determinados símbolos por otros según una regla fija.

El mayor desarrollo que tuvo el cifrado de mensajes, también conocido por criptografía, fue durante las dos guerras mundiales al establecer comunicaciones secretas militares y diplomáticas.

Cuando Shannon dio a conocer en 1949 la Teoría de las Comunicaciones Secretas, fue publicada por el NBS de Estados Unidos dando pauta al desarrollo del sistema criptográfico DES (Data Encryption Standards), la criptografía empezó a ser considerada como una ciencia aplicada debido a su relación con otras ciencias, como la estadística, la teoría de números, la teoría de la información y la complejidad computacional.

Se llama ***cifrado*** (encriptamiento) a la transformación que se realiza sobre el ***texto claro***[7], mediante una función parametrizada por una llave. La salida del proceso de cifrado se conoce como ***texto cifrado*** o ***criptograma***. El ***descifrar*** (descencriptar) también se le llama ***criptoanálisis***. Mientras que el arte de diseñar cifradores (conocido como ***criptografía***), y de descifrarlos se conoce colectivamente como ***criptología***. Vea la figura 4.1.

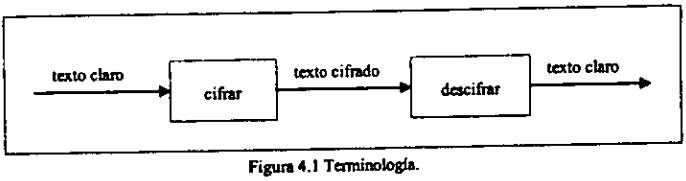


Figura 4.1 Terminología.

La raíz etimológica de la palabra criptología proviene de la palabra *kriptos*, que significa oculta y *graphos* que se traduce como escribir, lo que significa el arte de escribir mensajes en llave secreta o enigmáticamente.

Los cifrados digitales se pueden dividir en dos grupos, según la relación existente entre la llave de cifrado y la de descifrado. Cuando la llave de descifrado coincide con la de cifrado, se habla de ***cifrado simétrico*** o de ***llave secreta***. Por otra parte, si es imposible de obtener la llave de descifrado mediante la llave de cifrado, se trata de un ***cifrado asimétrico*** o de ***llave pública***.

4.3. Sistemas de llave secreta.

La criptografía de llave secreta utiliza algoritmos de cifrado muy complicados y rebuscados, valiéndose de operaciones como la transposición y la sustitución.

En este tipo de sistemas, el emisor E cifra el mensaje M con una determinada llave k :

$$E_k(M) = C$$

Y el receptor R podrá descifrar el mensaje M mediante la llave secreta k :

$$D_k(C) = M$$

En la figura 4.2. se visualiza lo descrito anteriormente.

[7] El término en inglés es *plain text*.

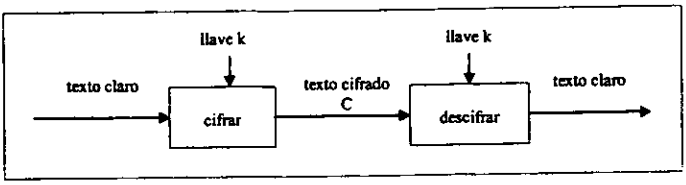


Figura 4.2. Sistemas de llave secreta.

Los cifrados de llave secreta se clasifican en dos grupos: los correspondientes a fuentes que generan *n* palabras llamados ***cifrados en bloques*** y los correspondientes a fuentes que generan letras denominados ***cifrados en flujos***.

4.3.1. Estándar de Cifrado de Datos (DES).

El cifrado más conocido es el llamado DES (Data Encryption Standard), catalogado como un cifrado en bloques, producto de transposiciones y sustituciones.

Las ideas de Shannon sobre la creación de nuevos sistemas de cifrado que consistían mediante la mezcla de operaciones como la transposición y la sustitución, fueron aprovechadas por IBM en los años sesenta, con la creación del sistema LUCIFER. En el año de 1976, este sistema fue adoptado por el gobierno de los Estados Unidos como estándar entre los sistemas de cifrado y lo denominó DES (Data Encryption Standard, Estándar de Cifrado de Datos).

El Estándar de Cifrado de Datos (DES), es un algoritmo simétrico: una sola llave es utilizada para cifrar y descifrar el mensaje. El texto normal se cifra en bloques de 64 bits y la salida que genera este algoritmo es un texto cifrado de 64 bits. El tamaño de la llave es de 56 bits (Usualmente es expresada por un número de 64 bits pero cada bit que se encuentra en la posición ocho o su múltiplo es utilizado para verificar la paridad, por lo que son ignorados los bits 8, 16,32,40, 48, 56 y 64). En la figura 4.3. se observa el esquema gráfico del algoritmo.

El primer paso en la implementación de este algoritmo es realizar la transposición (T_0) sobre el bloque de texto normal de 64 bits (BTN) a través de una permutación P, es decir, $T_0 = P(BTN)$. Después de pasar T_0 dieciséis veces por una función f , se transpone bajo la permutación inversa P^{-1}, obteniendo el bloque de texto cifrado. Las permutaciones P e P^{-1} se muestran en las siguientes matrices:

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	14	7

Tabla 4.1.Permutación inicial P.

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	22	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Tabla 4.2 Permutación final P^{-}

Estas matrices de permutación se leen de izquierda a derecha y de arriba hacia abajo, de tal manera que la matriz P transpone BTN = $d_1 \ldots d_{64}$ en $T_0 = d_{58}d_{50} \ldots d_7$, donde d_i representa el ***dígito*** que está en la posición i-ésima del bloque BTN.

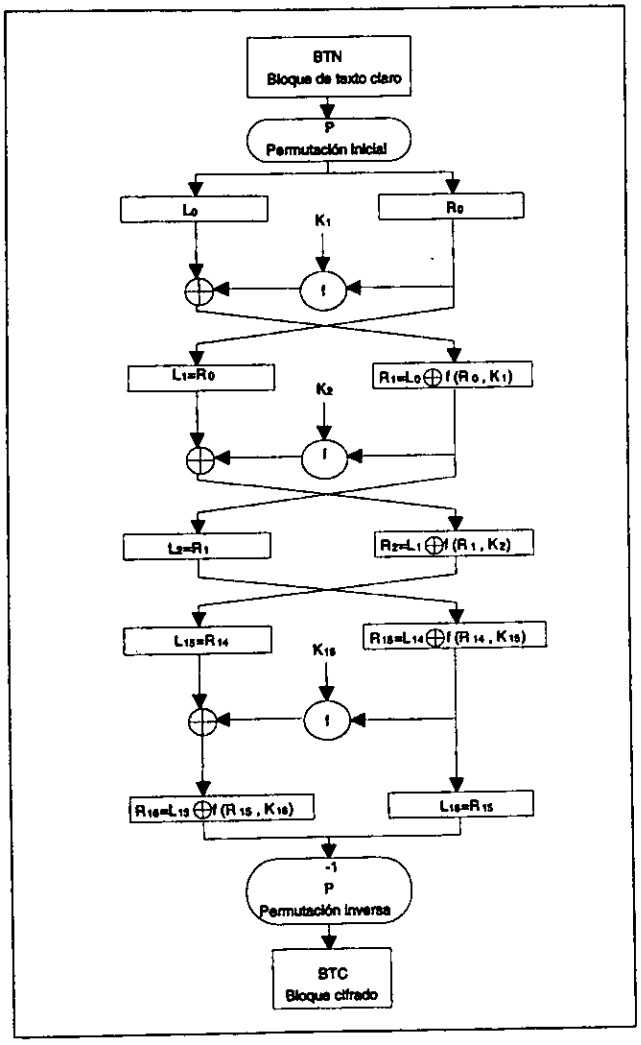


Figura 4.3. Estándar de cifrados de datos.

Después de la permutación inicial el bloque es dividido en un bloque izquierdo Li = $d_1 \ldots d_{32}$ y un bloque derecho Ri = $d_{33} \ldots d_{64}$, es decir, en dos mitades de 32 bits cada una. En el siguiente paso, el bloque de la derecha se convierte en

el bloque de la izquierda, mientras que el bloque de la derecha es el resultado de una operación OR EXCLISIVA entre el bloque izquierdo anterior y la función f, $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$, donde K_i es una llave de 48 bits. Este procedimiento se realiza *Ti* iteraciones, donde i = 1...16. Después de la última iteración, los bloques izquierdo y derecho de 32 bits no son intercambiadas con el fin de poder utilizar el mismo algoritmo para el descifrado.
Para realizar $f(R_{i-1}, K_i)$, mostrada en la figura 4.4, es necesario aumentar R_{i-1} de 32 a 48 bits, sin embargo se requiere ejecutar la operación $E(R_{i-1})$ usando la tabla 4.3 de selección de bits mostrada a continuación.

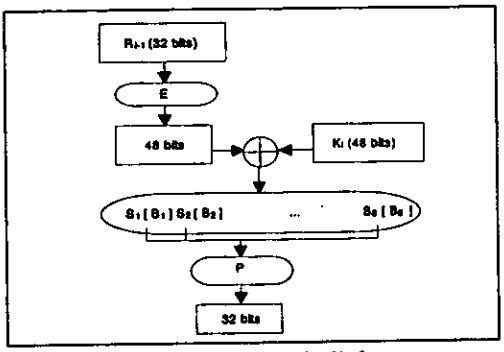


Figura 4.4. Proceso de la función f.

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Tabla 4.3. Selección de bits.

Posteriormente se realiza la operación OR EXCLUSIVA entre $E(R_{i-1})$ y K_i, y el resultado se divide en 8 bloques de 6 bits $B_1...B_8$. Con estos bloques se realizarán las operaciones de substitución de la siguiente manera. Las substituciones son realizadas mediante el empleo de 8 matrices llamadas cajas S. Cada caja S_i tiene como entrada 6 bits pertenecientes al bloque B_i y la salida $S_i(B_i)$ es de 4 bits. En la figura 4.5 se muestran las cajas S.

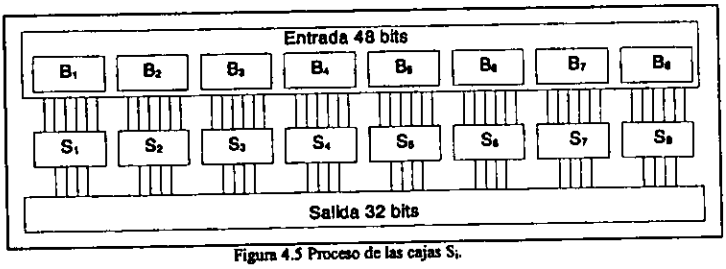


Figura 4.5 Proceso de las cajas S_i.

Cada caja *S* es una matriz de 4 filas y 16 columnas. La entrada de 6 bits del bloque B_i especifica el número de fila y columna de la caja *Si* para obtener la salida de 4 bits. Para describir con más detalle lo anteriormente mencionado, se ha considerado que la entrada de la caja S son los bits b_1, b_2, b_3, b_4, b_5, y b_6. Los bits b_1 y b_6 son combinados para formar un número de 2 bits, que indican la fila a seleccionar desde la 0 a la 3. Los otros bits b_2, b_3, b_4, y b_5 son combinados para formar un número de 4 bits, desde el 0 hasta el 15, los cuales corresponden a las columnas de la tabla. En la figura 4.6 se muestran las cajas *S*.

	Columnas																
Filas	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Caja
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	S_1
2	4	1	12	8	13	6	2	11	15	12	9	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	S_2
2	0	14	7	11	10	2	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S_3
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	S_4
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	S_5
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	8	0	13	3	4	14	5	11	
1	10	15	4	2	7	12	9	1	5	6	1	13	14	0	3	8	S_6
2	9	14	15	5	2	8	12	11	3	7	0	4	10	1	11	6	
3	4	3	2	12	9	5	15	7	10	11	14	1	7	6	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	S_7
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	S_8
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

Figura 4.6. Cajas *Si*.

Por ejemplo, si $B_1 = 011100$, entonces S_1 devuelve el valor que está en la fila 00=0 y columna 1110=14, que es 0 y se representa con 4 bits según 0000.

Los bloques $S_i(B_i)$ son concatenados y el bloque de 32 bits resultante es transpuesto según la permutación dada por la siguiente matriz PS

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Tabla 4.4 Matriz de permutación PS

El bloque devuelto por f (R_{i-1}, K_i) se denota como $P(S_1(B_1), \ldots, S_8(B_8))$.
Para determinar la seguridad de las cajas S deben satisfacer ciertas condiciones:

1. Las funciones booleanas utilizadas no deben ser lineales.
2. Debe existir un balance de ceros y unos.
3. No debe existir una correlación entre diferentes combinaciones de bits.
4. Los bits de salida deben comportarse de forma independiente con la entrada de cualquier bit complementado.

Una propiedad importante es el efecto avalancha, es decir, saber cuantos bits de salida de una caja S cambian cuando se cambia un subconjunto de bits de entrada. Es fácil imponer condiciones a las funciones booleanas para que satisfagan el criterio avalancha, pero construirlas es una difícil tarea. El criterio de avalancha estricto garantiza que exactamente la mitad de los bits de salida no cambian cuando un bit de entrada cambia.

Cálculo de la llave K_i

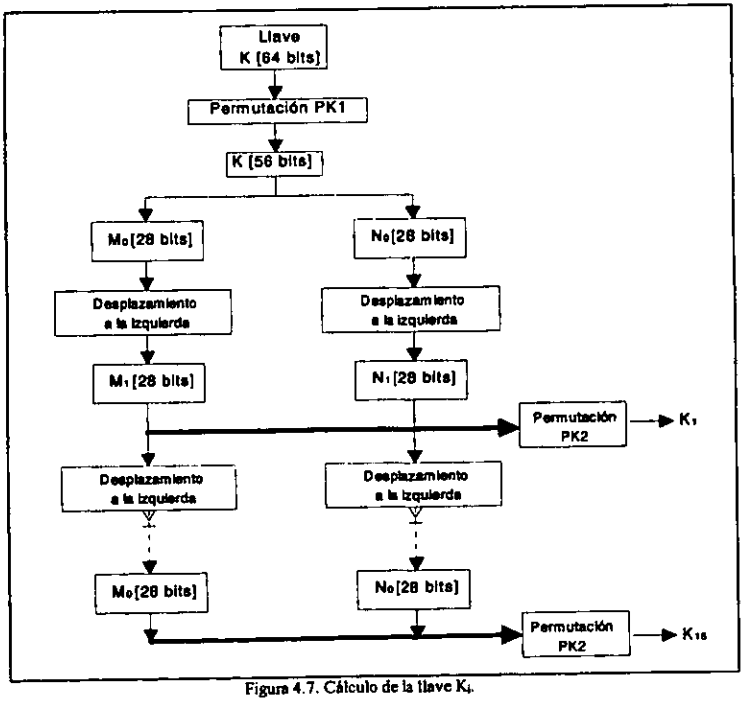


Figura 4.7. Cálculo de la llave K_i.

Como anteriormente se mencionó, la llave DES de 64 bits es reducida de tamaño. La llave es permutada por la matriz PK1 (Tabla 4.5) , obteniendo una llave de 56 bits.

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Tabla 4.5 Matriz de permutación PK1.

La llave de 56 bits es dividida en dos mitades M y N. Cada mitad es desplazada circularmente hacia la izquierda por uno o dos bits, dependiendo de la iteración. Este cambio se muestra en la tabla 4.6.

Iteración	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Desplazamiento	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Tabla 4.6. Desplazamiento de bits para cada M y N por iteración.

Si se denota M_i y N_i a los valores de M y N usados en cada iteración para obtener K_i, se tiene que $M_i = DI_i(M_{i-1})$ y $N_i = DI_i(N_{i-1})$, donde DI_i es un desplazamiento circular a la izquierda de un número de posiciones dado por la iteración correspondiente. Por ejemplo:

$$DI_2(M_1) = DI_2(10110101011100011011111101010) = (01101010111000110111110101 01) = M_2$$
$$DI_3(M_2) = DI_3(01101010111000110111110101 01) = (10101011100011011111010101 01) = M_3$$

La llave K_i es el resultado de la transposición $K_i = PK2(M_i * N_i)$, donde PK2 es la permutación dada por la matriz siguiente:

14	17	11	24	1	5
3	29	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Tabla 4.7. Matriz de permutación PK2.

La matriz de permutación PK2 reduce el tamaño del bloque de 56 a 48 dígitos, ya que se eliminan las posiciones 9,18,22,25,35,43 y 54.

Descifrado

El proceso de descifrado se realiza con el mismo algoritmo utilizado para el cifrado pero teniendo en cuenta que la permutación inicial se hará con los bloques L_{16} y R_{16}, siendo L_0 y R_0 los bloques obtenidos en el último paso. La llave que se utilizará en la primer iteración es K_{16}, luego la K_{15}, y así sucesivamente hasta K_1 en la decimosexta iteración. Por lo tanto, las expresiones que definen el cálculo realizado en cada paso serán las siguientes:

$$R_{i-1} = L_i,$$

$$L_{i-1} = R_i \oplus f(L_{i-1}, K_i)$$

Seguridad del algoritmo

DES se basó en un cifrado de IBM , llamado LUCIFER, que usaba una llave de 128 bits. Cuando el gobierno de los Estados Unidos quiso estandarizar un método de cifrado para uso no confidencial, invitó a IBM a debatir el asunto con la NSA (National Security Agency, Agencia Nacional de Seguridad), dependencia de descifrados del gobierno. El resultado de estos debates fue que la reducción de la llave de 128 a 56 bits del algoritmo de IBM y decidió mantener en secreto el proceso de diseño del DES.

La única forma conocida de violar el algoritmo es probar a descifrar la información con todas las posibles llaves. Puesto que constan de 56 bits habría que probar con 2^{56}, es decir, 72, 057, 594, 037, 927, 936 llaves distintas, esto haría pensar que encontrar la llave se llevaría años. Sin embargo, en 1997, dos investigadores de criptografía Diffie y Hellman diseñaron una máquina para violar el DES y estimaron que podría construirse por unos 20 millones de dólares. Esta máquina podría encontrar la llave mediante una búsqueda entre 2^{56} llaves posibles en menos de un día. En 1994 se presentó un diseño detallado de una máquina que puede descifrar el DES en unas diez horas.

4.3.1.1. Triple DES.

Al ver que DES se convertía en un sistema vulnerable debido al uso de llaves con una longitud muy corta se empezaron a diseñar algunas variantes para aumentar su efectividad. Una opción fue el diseño de Triple-DES mediante la triple codificación (Figura 4.8). Su forma de operar es en tres etapas y requiere el uso de tres llaves: la primera llave *K1* es utilizada por el emisor *E* para cifrar el mensaje *M*, después es descifrado por una segunda llave *K2* y finalmente, vuelve a ser cifrado con una tercera llave *K3*.

$$C = E_{K3}(D_{K2}(E_{K1}(M)))$$

El receptor *R* descifra el mensaje con la llave *K3*, el resultado obtenido lo cifra con la llave *K2* para ser descifrado con la llave *K1*, y así obtener el mensaje original.

$$M = R_{K3}(D_{K2}(R_{K1}(C)))$$

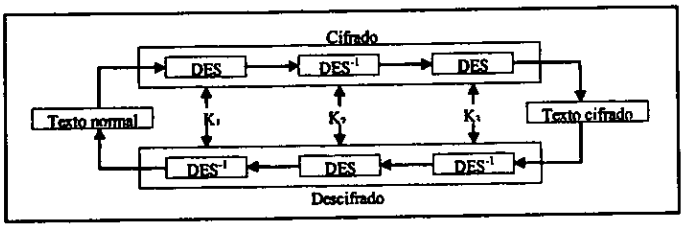


Figura 4.8. Triple-DES.

Para violar este sistema de seguridad se tendría que realizar 2^{168} intentos (en comparación con 2^{56} del algoritmo original), para obtener la llave de cifrado.

4.3.1.2. Criptoanálisis diferencial.

Existen métodos para poder atacar los cifrados en bloques, uno de ellos es el criptoanálisis diferencial diseñado por Biham y Shamir en 1990. Esta técnica funciona comenzando por un par de bloques de texto normal que difieren en una cantidad pequeña de bits. Esta diferencia es analizada en cada iteración interna a medida que avanza el cifrado. Usando las diferencias en el texto cifrado resultante, se asigna probabilidades diferentes a las llaves utilizadas. A medida que se analiza el par de textos cifrados, la llave que tenga una mayor probabilidad será la llave del cifrado.
El procedimiento del criptoanálisis diferencial en el algoritmo DES es el siguiente. En la función f, mostrada en la figura 4.9, consideremos un par de textos normales X y X', cuya diferencia es ΔX. Las respectivas salidas de la función f son Y y Y', conocidas al igual que su diferencia ΔY. También se tiene como dato la permutación de expansión E y la permutación P, por lo que las diferencias ΔA y ΔC son conocidas. Por otra parte, los valores de B y B' no son conocidos, pero su diferencia ΔB sí, y es igualada a ΔA:

$$\Delta B = (A \oplus Ki) \oplus (A' \oplus Ki) = \Delta A$$

No todos los valores de ΔC son igualmente parecidos. La combinación de ΔA y ΔC permite obtener valores para los bits de A OR EXLCUSIVA *Ki*, y A' OR EXLCUSIVA *Ki*. Ya que el valor de A y A' se sabe, puede obtenerse la información acerca de *Ki*.

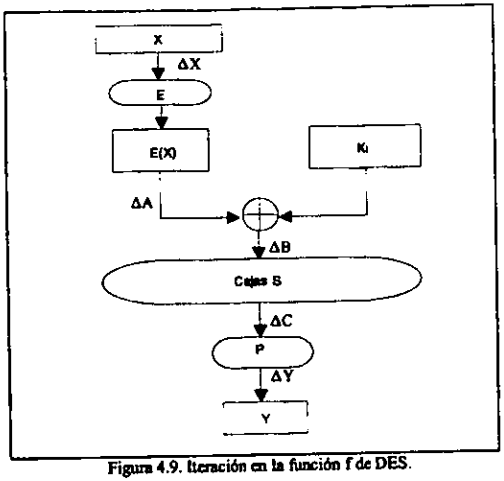


Figura 4.9. Iteración en la función f de DES.

Las diferencias entre un par de textos normales causan diferencias en el par de textos cifrados. Estas diferencias son llamadas características y se extienden sobre el número de iteraciones del algoritmo. Por lo tanto, existe una diferencia de entrada, una diferencia en cada iteración y una diferencia de salida, con una probabilidad específica. Se pueden encontrar estas características generando una tabla, donde las filas representan la posible entrada de la operación OR EXCLUSIVA, las columnas son la salida de la operación OR EXCLUSIVA, y las entradas representan el número de veces que ocurre una salida OR EXLUSIVA, a partir de una entrada OR EXCLUSIVA dada. Se puede generar una tabla para cada una de las 8 cajas S de DES.

Por ejemplo en la figura 4.10(a), se muestra una iteración para encontrar sus características. La diferencia de entrada del lado izquierdo es L. La diferencia de salida del lado derecho es 0. Como la diferencia que entra en la función f es nula, entonces no hay una diferencia de salida de esta función; por lo que, la diferencia de salida del lado izquierdo es $L \oplus 0 = L$, y la diferencia de salida del lado derecho es 0. Esto es una característica trivial, y es cierta con una probabilidad de 1.

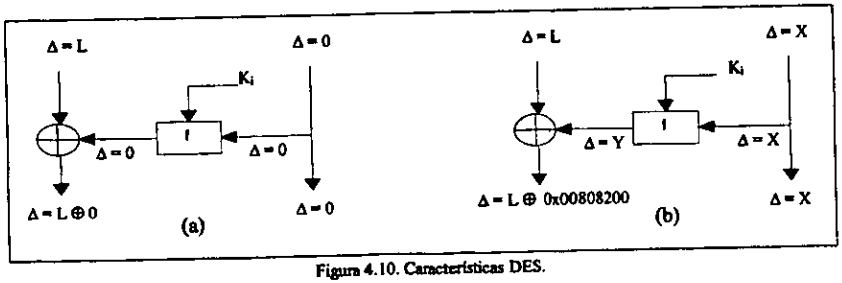


Figura 4.10. Características DES.

En la figura 4.10(b), la diferencia de entrada del lado izquierdo es L y la diferencia de entrada del lado derecho es 0x60000000. Con una probabilidad de 14/64, la diferencia de salida de la función f es L ⊕ 0x00808200. Esto significa que la diferencia de salida del lado izquierdo es L ⊕ 0x00808200 y la diferencia de salida del lado derecho es 0x60000000, con una probabilidad de 14/64.

Las anteriores características pueden ser unidas. Y asumiendo que las iteraciones son independientes, las probabilidades pueden ser multiplicadas. En la figura 4.11 se juntan las dos características descritas. La diferencia de entrada del lado izquierdo es 0x00808200 y diferencia de entrada del lado derecho es 0x60000000. Al final de la primera iteración, la diferencia de entrada del lado izquierdo y la diferencia de salida de la función f se cancelan, dejando una diferencia de 0. Entonces, para la segunda iteración, la entrada del lado izquierdo es 0x60000000 y la entrada del lado derecho es 0. Esta segunda iteración tiene una probabilidad de 14/64.

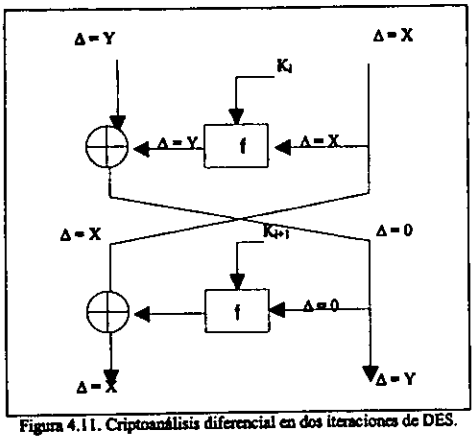


Figura 4.11. Criptoanálisis diferencial en dos iteraciones de DES.

Un par de texto que satisface las características es un par correcto y sugerirá la llave correcta de la iteración. Un par de texto normal el cual no satisface es un par incorrecto y sugerirá una llave aleatoria de la iteración.

Para encontrar la llave correcta de la iteración se tiene que realizar muchos cálculos, de tal manera, que una llave resulte ser la más frecuente de todas las otras.

El ataque diferencial sobre la iteración *n* de DES recobrará la llave de 48 bits usada en esa iteración y el resto de los 8 bits de la llave son obtenidos adivinando.

En la tabla 4.8 se muestra un sumario de los mejores ataques diferenciales en contra de DES con un número variado de iteraciones. La primera columna es el número de iteraciones. Las siguientes dos columnas son el número de textos normales escogidos o textos normales conocidos que deben ser examinados para el ataque, y la cuarta columna es el número de textos normales actualmente analizados. La última columna se refiere a la complejidad del análisis.

No de iteraciones	Textos normales seleccionados	Textos normales conocidos	Textos normales analizados	Complejidad del análisis
8	2^{14}	2^{18}	4	2^{9}
9	2^{24}	2^{44}	2	2^{32}
10	2^{24}	2^{43}	2^{14}	2^{15}
11	2^{31}	2^{47}	2	2^{32}
12	2^{31}	2^{47}	2^{21}	2^{21}
13	2^{39}	2^{52}	2	2^{32}
14	2^{39}	2^{51}	2^{29}	2^{29}
15	2^{47}	2^{56}	2^{7}	2^{37}
16	2^{47}	2^{55}	2^{36}	2^{37}

Tabla 4.8..Ataques de criptoanálisis diferencia contra DES.

El mejor ataque en contra de las 16 iteraciones DES requiere escoger 2^{47} textos normales. La resistencia de DES puede ser mejorada incrementado el número de iteraciones. Si el número de iteraciones son 19 ó más, el criptoanálisis diferencial llega a ser imposible porque se requiere más de 2^{64} posibles bloques de texto normal.

4.3.3 Algoritmo Internacional de Cifrado de Datos (IDEA).

El algoritmo de seguridad IDEA fue desarrollado a partir de PES (Proposed Encryption Standard, Estándar de Cifrado Propuesto), diseñado por Xuejia Lai y James Massey. Al publicar Biham y Shamir el criptoanálisis diferencial, el cifrado PES fue reforzado en contra de ataques y lo llamaron IPES (Improved Proposed Encryption Algorithm, Algoritmo de Cifrado Mejorado Propuesto). Posteriormente, IPES cambió su nombre por IDEA en 1992. IDEA opera sobre bloques de texto normal de 64 bits. Usa tres operaciones algebraicas: OR EXCLUSIVO, suma módulo 2^{16} y multiplicación módulo $2^{16}+1$, todas sobre números de 16 bits. El tamaño de la llave *k* es de 128 bits y es usada para generar 52 subllaves ($Z_i^{(r)}$), cada una de ellas de 16 bits, 6 por cada una de las ocho iteraciones y 4 para la transformación final. El mismo algoritmo es usado para cifrar y descifrar la información. A continuación se describe los pasos del algoritmo que se muestran en la tabla 4.9.

Operación	Cálculo
1	Multiplicar X_1 y la primera subllave.
2	Sumar X_2 y la segunda subllave.
3	Sumar X_3 y la tercera subllave.
4	Multiplicar X_4 y la cuarta subllave.
5	Aplicar OR EXCLUSIVO entre el resultado del paso (1) y (3).
6	Aplicar OR EXCLUSIVO entre el resultado del paso (2) y (4).
7	Multiplicar el resultado del paso (5) con la quinta subllave.
8	Sumar el resultado del paso (6) y (7).
9	Multiplicar el resultado del paso (8) con la sexta subllave.
10	Sumar el resultado del paso (7) y (9).
11	Aplicar OR EXCLUSIVO entre el resultado del paso (1) y (9).
12	Aplicar OR EXCLUSIVO entre el resultado del paso (3) y (9).
13	Aplicar OR EXCLUSIVO entre el resultado del paso (2) y (10).
14	Aplicar OR EXCLUSIVO entre el resultado del paso (4) y (10).

Tabla 4.9. Operaciones en cada iteración en IDEA.

Primero el bloque de 64 bits de texto normal es dividido en 4 bloques de 16 bits X_1, X_2, X_3, y X_4. Estos cuatro bloques son la entrada de la primera iteración del algoritmo. Existen 8 iteraciones en total. En cada iteración los cuatro bloques pasan por las operaciones OR EXCLUSIVO, suma y multiplicación junto con las llaves de 16 bits correspondientes.

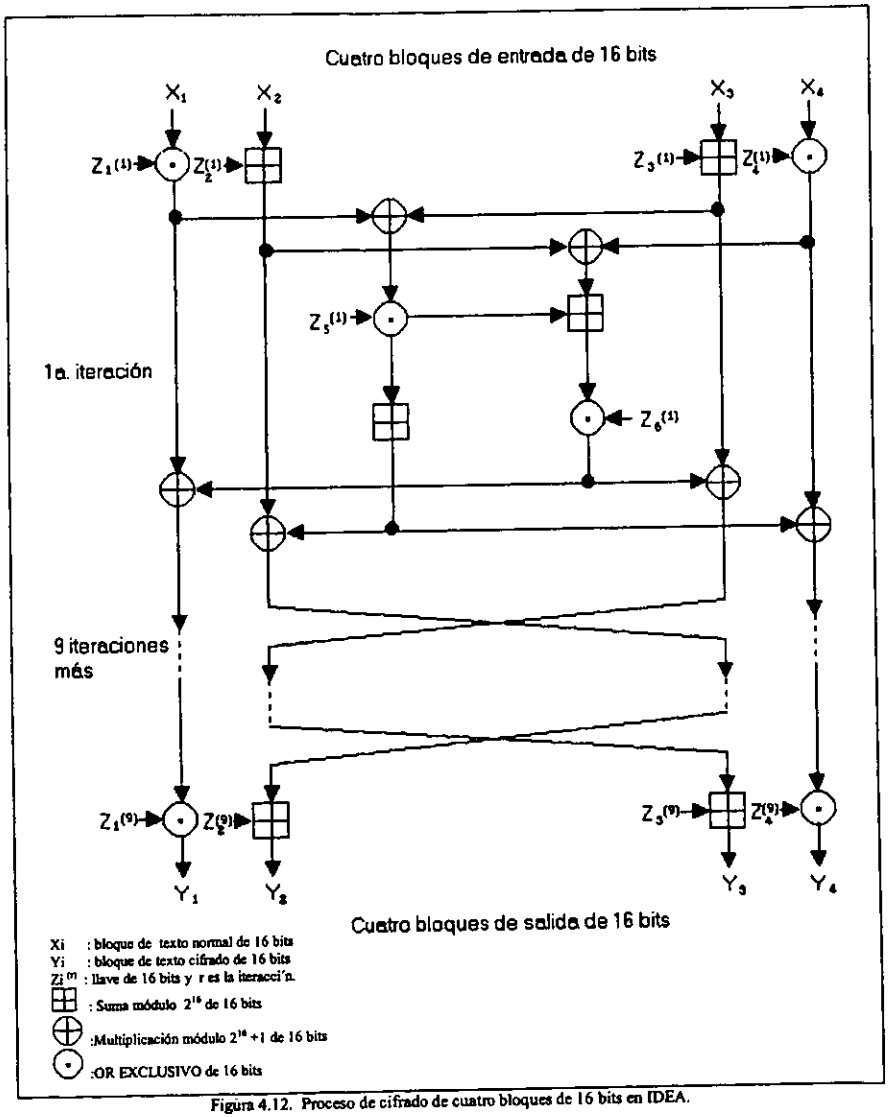


Figura 4.12. Proceso de cifrado de cuatro bloques de 16 bits en IDEA.

Finalmente se obtiene un bloque cifrado de 64 bits como resultado de la combinación de los cuatro bloques con las cuatro últimas subllaves $Z_1(9)$, $Z_2(9)$, $Z_3(9)$, y $Z_4(9)$.

En la novena iteración del algoritmo sólo se realizan las primeras cuatro operaciones descritas en la tabla anterior, junto con las subllaves correspondientes a esa iteración. El resultado de estas operaciones es el bloque cifrado. Finalmente, estos cuatro bloques son insertados para producir el texto cifrado de 64 bits. En la figura 4.12 se observan las operaciones que se realizan en cada iteración del algoritmo de IDEA.

Subllaves

Para generar las llaves $Z_i^{(r)}$ utilizadas en cada iteración, se divide la llave de 128 bits en 8 llaves de 16 bits. De estas llaves, las primeras 6 son utilizadas para la primera iteración y las otras 2 en la segunda iteración (Figura 4.13). Posteriormente, la llave K es rotada 25 bits a la izquierda y de nuevo es dividida en ocho. Las primeras cuatro subllaves son usadas en la segunda iteración y las otras cuatro en la tercera iteración. La llave es de nuevo rotada 25 bits hacia la izquierda para las siguientes 8 llaves de 16 bits. El proceso es el mismo para cada iteración hasta el termino del algoritmo.

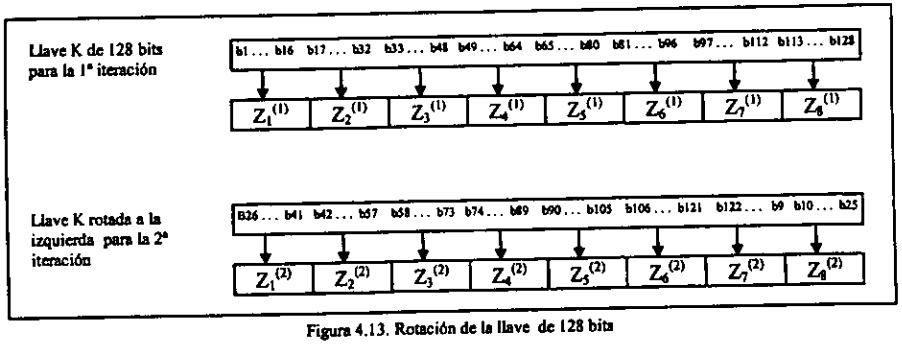


Figura 4.13. Rotación de la llave de 128 bits

Para el descifrado se aplica el mismo algoritmo, excepto que ahora, las llaves de cifrado serán invertidas de orden, así como también, les será aplicado el inverso aditivo o el inverso multiplicativo a cada una. Al inicio del descifrado se utilizarán las ultimas llaves de cifrado $Z_1^{(9)}$, $Z_2^{(9)}$, $Z_3^{(9)}$ y $Z_4^{(9)}$. El proceso es el siguiente: la primera y cuarta llave de descifrado son igual al inverso de $Z_1^{(9)}$ y $Z_4^{(9)}$ módulo $2^{16}+1$, respectivamente. Mientras que la segunda y tercera llave son el inverso aditivos de $Z_2^{(9)}$ y $Z_3^{(9)}$ módulo $2^{16}+1$. En la siguiente tabla se muestran las llaves de cifrado y descifrado.

Iteración	Llaves de cifrado						Llaves de descifrado					
1	$Z_1^{(1)}$	$Z_2^{(1)}$	$Z_3^{(1)}$	$Z_4^{(1)}$	$Z_5^{(1)}$	$Z_6^{(1)}$	$Z_1^{(9)}{}^{-1}$	$-Z_2^{(9)}$	$-Z_3^{(9)}$	$Z_4^{(9)}{}^{-1}$	$Z_5^{(8)}$	$Z_6^{(8)}$
2	$Z_1^{(2)}$	$Z_2^{(2)}$	$Z_3^{(2)}$	$Z_4^{(2)}$	$Z_5^{(2)}$	$Z_6^{(2)}$	$Z_1^{(8)}{}^{-1}$	$-Z_2^{(8)}$	$-Z_3^{(8)}$	$Z_4^{(8)}{}^{-1}$	$Z_5^{(7)}$	$Z_6^{(7)}$
3	$Z_1^{(3)}$	$Z_2^{(3)}$	$Z_3^{(3)}$	$Z_4^{(3)}$	$Z_5^{(3)}$	$Z_6^{(3)}$	$Z_1^{(7)}{}^{-1}$	$-Z_2^{(7)}$	$-Z_3^{(7)}$	$Z_4^{(7)}{}^{-1}$	$Z_5^{(6)}$	$Z_6^{(6)}$
4	$Z_1^{(4)}$	$Z_2^{(4)}$	$Z_3^{(4)}$	$Z_4^{(4)}$	$Z_5^{(4)}$	$Z_6^{(4)}$	$Z_1^{(6)}{}^{-1}$	$-Z_2^{(6)}$	$-Z_3^{(6)}$	$Z_4^{(6)}{}^{-1}$	$Z_5^{(5)}$	$Z_6^{(5)}$
5	$Z_1^{(5)}$	$Z_2^{(5)}$	$Z_3^{(5)}$	$Z_4^{(5)}$	$Z_5^{(5)}$	$Z_6^{(5)}$	$Z_1^{(5)}{}^{-1}$	$-Z_2^{(5)}$	$-Z_3^{(5)}$	$Z_4^{(5)}{}^{-1}$	$Z_5^{(4)}$	$Z_6^{(4)}$
6	$Z_1^{(6)}$	$Z_2^{(6)}$	$Z_3^{(6)}$	$Z_4^{(6)}$	$Z_5^{(6)}$	$Z_6^{(6)}$	$Z_1^{(4)}{}^{-1}$	$-Z_2^{(4)}$	$-Z_3^{(4)}$	$Z_4^{(4)}{}^{-1}$	$Z_5^{(3)}$	$Z_6^{(3)}$
7	$Z_1^{(7)}$	$Z_2^{(7)}$	$Z_3^{(7)}$	$Z_4^{(7)}$	$Z_5^{(7)}$	$Z_6^{(7)}$	$Z_1^{(3)}{}^{-1}$	$-Z_2^{(3)}$	$-Z_3^{(3)}$	$Z_4^{(3)}{}^{-1}$	$Z_5^{(2)}$	$Z_6^{(2)}$
8	$Z_1^{(8)}$	$Z_2^{(8)}$	$Z_3^{(8)}$	$Z_4^{(8)}$	$Z_5^{(8)}$	$Z_6^{(8)}$	$Z_1^{(2)}{}^{-1}$	$-Z_2^{(2)}$	$-Z_3^{(2)}$	$Z_4^{(2)}{}^{-1}$	$Z_5^{(1)}$	$Z_6^{(1)}$
salida	$Z_1^{(9)}$	$Z_2^{(9)}$	$Z_3^{(9)}$	$Z_4^{(9)}$			$Z_1^{(1)}{}^{-1}$	$-Z_2^{(1)}$	$-Z_3^{(1)}$	$Z_4^{(1)}{}^{-1}$		

Tabla 4.10. Llaves de cifrado y descifrado de IDEA

Seguridad

La llave que se utiliza en IDEA tiene una longitud de 128 bits (dos veces más grande que DES), por lo que se requerirá 10^{38} intentos para violar el sistema. Este algoritmo se diseñó para resistir el criptoanálisis diferencial.

Willi Meier examinó las tres operaciones algebraicas de IDEA y señaló que mientras éstas fueran independientes entre sí, existen formas para ser simplificadas, de tal manera que se facilite el criptoanálisis y lo disminuya en un porcentaje de tiempo. El ataque del criptoanálisis es más eficiente que la fuerza bruta para 2 iteraciones de IDEA (2^{42} operaciones), pero menos eficiente para 3 iteraciones IDEA o más. Normalmente IDEA con 8 iteraciones está a salvo. Actualmente las implantaciones de software mediante IDEA son dos veces más rápidas que DES. IDEA encripta datos a 0.888 Mbs en una máquina de 33 MHz, y 2.4 Mbs en una máquina 486 a 66 MHz.

4.4. Cifrado en flujo.

El cifrado en flujo consiste en combinar textos formados por letras con un flujo de bits secretos. El primero en proponer un cifrado en flujo fue Vernam. El cifrado de Vernam se basa en combinar el texto normal con el flujo de bits secretos mediante el operador OR EXCLUSIVA (suma módulo 2). La secuencia cifrante se produce mediante un generador de bits. Vea figura 4.16.

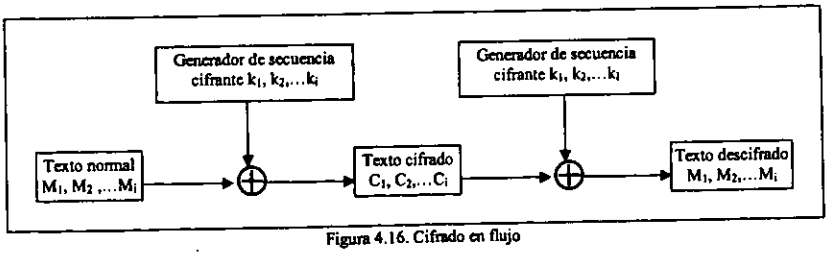


Figura 4.16. Cifrado en flujo

Si se tiene un flujo de texto igual a $M_1, M_2, \ldots Mi$, para producir un cifrado en flujo $C_1, C_2, \ldots C_i$, se tiene que:

$$C_i = M_i \oplus k_i, \qquad \text{donde } k_i \text{ es el flujo de bits secretos}$$

Para descifrar se realiza la operación OR EXCLUSIVA entre el cifrado en flujo y el mismo flujo de bits secretos:

$$M_i = C_i \oplus k_i ,$$

Es decir,

$$M_i \oplus k_i \oplus k_i = M_i$$

Los cifrados en flujo pueden dividirse en cifrados síncronos y cifrados autosíncronos. Los cifrados síncronos son aquellos en los que cada estado sólo depende del anterior y no de los caracteres de la secuencia del mensaje original. Esto implica que si durante la transmisión se pierde un símbolo del criptograma, entonces se pierde la sincronización entre emisor y receptor, por lo que ambos tienen que sincronizar sus generadores de llave antes de continuar.

Los cifrados autosíncronos se caracterizan porque cada símbolo de texto cifrado depende de un número fijo de símbolos del texto cifrado anterior. Esto elimina la necesidad de sincronización, pero un error de transmisión o pérdida de un carácter origina la pérdida o mala obtención de un número de símbolos del mensaje original.

Si la secuencia cifrante deber ser la misma para el emisor y el receptor se debe implantar en el generador el mismo algoritmo para producir la misma secuencia. Esta secuencia debe ser seudoaleatoria para que el intruso no pueda reproducirla, pero los usuarios que tengan la llave sí puedan.

La llave debe cambiar en el cifrado de cada nuevo texto, para esto se requiere de un parámetro llamado vector de inicialización o semilla , el cual, permite generar una nueva secuencia sin modificar el algoritmo del generador.

Un método de generación de secuencias seudoaleatorias es el conocido como Registro de Desplazamiento con Realimentación Lineal (RDRL). Este generador produce una secuencia en la que cada término depende de forma lineal de sus predecesores.

Las secuencias obtenidas mediante determinados RDRL tienen un período bastante grande y son fáciles de implementar. Sin embargo, el conocimiento de una parte de la secuencia cifrante permite al intruso descubrir al generador y toda la secuencia. Existen funciones más complejas que no son lineales y que generan secuencias criptográficas más resistentes como son los combinadores no lineales.

4.5. Sistemas de llave pública

La parte débil de la mayoría de los sistemas de llave secreta es el problema de distribución de llaves. Sin importar lo robusto que sea el criptosistema, si algún intruso encuentra la llave de cifrado ya no valdrá nada el sistema de seguridad, debido a que dicha llave también es utilizada para descifrar (o bien de ella se podía derivar fácilmente la otra).

En 1976, dos ingenieros electrónicos de la Universidad de Stanford, Whitfield Diffie y Martin Hellman, sugieren una clase nueva de criptosistema seguros, en el que las llaves de cifrado y descifrado son diferentes, y además, la llave de descifrado no puede derivarse de la llave de cifrado. La idea consiste en usar una transformación criptográfica T_k de

fácil aplicación, no inversible sin información adicional; de tal forma, que a partir de ella sea difícil de hallar la transformación inversa T_k^{-1}. En este esquema, para determinar la función T_k se requiere de una llave de cifrado k (llave pública), y una llave de descifrado d (llave secreta o privada), para realizar el cálculo de la inversa T_k^{-1}.
El método funciona como sigue. Como anteriormente se mencionó, los sistemas de llave pública utilizan dos llaves, una pública k y otra privada d. Básicamente, el emisor E cifra la información I con la llave pública k del receptor R, que deberá ser conocida:

$$\text{información encriptada por el emisor} = k_R(I_E)$$

Y el receptor R, descifra la información con su propia llave privada d, sólo conocida por él y que debe permanecer en secreto:

$$\text{información descifrada} = d_R(I_E)$$

El cifrador del mensaje extrae la llave pública del destinatario accediendo a una base de datos con las llaves públicas de los posibles destinatarios.
Por lo tanto, la criptografía de llave pública requiere que cada usuario tenga dos llaves: una llave pública usada por los demás para cifrar mensajes a enviar a ese usuario, y una llave privada, que necesita el usuario para descifrar los mensajes.

4.5.1. RSA.

El algoritmo RSA fue desarrollado en el año de 1977 por Ronald Rivest, Adi Shamir y Leonard Adleman, y está basado en el problema de hallar los factores primos de grandes números.
En primer lugar, se eligen dos números primos grandes, p y q, y se calcula:

$$n = p \cdot q$$

Se escoge un número e que no tenga divisores en común, aparte de 1, con $(p-1) \cdot (q-1)$; es decir, el máximo común divisor entre e y $(p-1) \cdot (q-1)$ debe ser 1, siendo recomendable que d sea mayor que p y q, y menor que n:

$$n > e > p \quad , \quad e > p$$

y se calcula un número d tal que

$$(e \cdot d) = 1 \bmod ((p-1) \cdot (q-1))$$

es decir,

$$d = e^{-1} \bmod ((p-1) \cdot (q-1))$$

siendo *mod* el operador módulo o resto de la división entera.

Los usuarios publican los valores *e* y *n*, que constituirán su llave pública y que todos los posibles remitentes de mensajes deben conocer, mientras que *d* deberá permanecer en secreto y formará junto con *n*, la llave privada de cada usuario. También deberán permanecer en secreto *p* y *q* que se utilizarán sólo para la obtención de las llaves.
Cualquier usuario que quiera enviar información la cifrará utilizando la función:

$$C = M^e \bmod n$$

donde *M* es el mensaje y *C* el mensaje cifrado que se obtendrá.
Para cifrar un texto, es necesario previamente codificar el texto *M* en un sistema numérico decimal o binario, y dividir en bloques m_i de tamaño *j* o *j-1* de forma que, según sea el alfabeto usado el decimal o el binario cumpla en cada caso $10^{j-1} < n < 10^j$ ó $2^{j-1} < n < 2^j$. Por ejemplo, si M = m1 m2 m3 ... se hará:

$$c1 = m1^e \bmod n$$
$$c2 = m2^e \bmod n$$
$$c3 = m3^e \bmod n \ldots$$

Obteniéndose un conjunto de bloques con la información cifrada, C = c1 c2 c3 ... que se enviarán al destinatario del mensaje.
Para descifrar el mensaje, el receptor empleará la función:

$$M = C^d \bmod n$$

es decir, para cada bloque recibido

$$m1 = c1^d \bmod n$$
$$m2 = c2^d \bmod n$$
$$m3 = c3^d \bmod n \ldots$$

Obteniendo los bloques de información *mi*, que sólo hará falta transformarlos en los correspondientes caracteres para tener el mensaje completo *M*. Se observa que es necesario tener llave privada *d* o los valores *p* y *q* para descifrar.

Seguridad

Entre los posibles ataques al algoritmo RSA están la fuerza bruta y los métodos matemáticos La fuerza bruta involucra tratar todas las posibles combinaciones de números para obtener la llave privada, y su complejidad de búsqueda aumenta cuando la llave privada y la pública son muy grandes. Claro que entre más grandes sean las llaves más lento será el cómputo.
Los métodos mediante ataques matemáticos consisten principalmente en la factorización. Si se encuentran dos números *(p,q)*, de tal forma que $n = p.\ q$ se puede encontrar el valor de *e* y *d*. Para encontrar esos dos números primos se utiliza la función ϕ de Euler, en la cual

$$\phi(n) = \phi(p).\ \phi(q) = (p-1)(q-1)$$

En la siguiente tabla podemos encontrar algunos valores de la función de Euler $\phi(n)$:

n	$\phi(n)$	n	$\phi(n)$	n	$\phi(n)$	n	$\phi(n)$	n	$\phi(n)$
1	1	6	10	11	10	16	8	21	12
2	1	7	4	12	4	17	16	22	10
3	2	8	12	13	12	18	6	23	22
4	2	9	6	14	6	19	18	24	8
5	4	10	8	15	8	20	8	25	20

Tabla 4.16. Valores de la función de Euler.

Por ejemplo, si $n = 21$ entonces $\phi(21) = 12 = \phi(3) . \phi(7) = 2.6 = (3-1).(7-1)$, donde $p=3$ y $q=7$.

Los laboratorios de RSA han empleado llaves de 100,110,120 dígitos. El último desafío es el RSA-130 con una llave de 130 dígitos. La tabla 4.17, muestra el tamaño de dígitos y bits, así como el número de instrucciones necesarias para romper la llave y el algoritmo que se utilizó. La mediada MIPS por año se define como un millón de instrucciones por segundo ejecutadas en un procesador durante un año. Una pentinum a 200 MHz es una máquina de 50 MIPS.

Número de dígitos	Número de bits	Fecha de logro	MIPS por año	Algoritmo
100	332	1991	7	Criba cuadrática
110	365	1992	75	Criba cuadrática
120	398	1994	830	Criba cuadrática
130	431	1996	500	Criba de campos numéricos generalizado
140	465	1999	700	Criba de campos numéricos generalizado

Tabla 4.17. Factorización de llaves.

En la actualidad la factorización se está realizando con más facilidad y esto es porque los ordenadores son cada vez más rápidos e interconectados funcionan mejor, y los algoritmos de factorización cada día son más eficientes. Por el momento, el reto de RSA es tener una llave de 155 cifras, es decir, de 512 bits.

4.6. Gestión de Llaves.

En los sistemas de llave pública, la distribución de llaves ya no es un problema, cada usuario debe mantener su propia llave privada y la llave pública se almacena en una base de datos o directorio de llaves públicas, de tal manera que cuando los usuarios cifran la información utilizan la llave pública del destinatario que extraerán del directorio de llaves públicas.

Sin embargo, en la distribución de llaves de los sistemas de llave privada surgen algunos problemas, principalmente por el número de llaves que se deben gestionar. Un método utilizado consiste en disponer de un centro de almacenamiento y distribución de llave donde el usuario sólo necesita una llave para utilizar sus servicios. Con este sistema, si un usuario desea comunicarse con otro, envía un mensaje a ese centro, cifrado con la llave que posee y comparte con el centro, en el que solicita una llave para ello. El centro generará una llave y se la enviará a los

usuarios que van a establecer la comunicación, cifrada con la llave de cada uno. Tras descifrar la llave, ambos usuarios la utilizarán para cifrar la información que se intercambien, teniendo que solicitar una nueva llave cada vez que deseen comunicarse.

Para la distribución de llaves en sistemas de llave privada también se puede emplear un sistema de llave pública, tal como el algoritmo de Diffie-Hellman para intercambio de llaves.

4.6.1. Algoritmo de Diffie-Hellman.

En 1976, Whit Diffie y Martin Hellman desarrollaron un algoritmo para la distribución de llaves que hace posible establecer una llave secreta mediante el intercambio de mensajes cuya intercepción no permitirá deducir la llave. En la actualidad es un método muy empleado para el intercambio de llaves y su funcionamiento es como se indica a continuación.

Suponiendo que dos usuarios A y B desean obtener una llave común, los pasos que deben seguir son:

- Establecer un número primo grande *p* y un entero *g*, que pueden ser públicos e incluso compartidos por otros usuarios.
- Cada uno elige un número positivo menor que *p-2* que mantendrán en secreto. Siendo *a* el número elegido por el usuario A y *b* el elegido por el usuario B proceden a calcular:

$$\text{Usuario A: } X_A = g^a \bmod p$$
$$\text{Usuario B: } X_B = g^b \bmod p$$

- Se intercambian los valores obtenidos, es decir, el usuario A envía X_A al usuario B y el usuario B envía X_B al usuario A y con los valores recibidos calculan:

$$\text{Usuario A: } Y_A = X_B\, g^a \bmod p$$
$$\text{Usuario B: } Y_B = X_A\, g^b \bmod p$$

al sustituir en estas ecuaciones X_A y X_B por la ecuaciones anteriores quedará:

$$Y_A = g^{ba} \bmod p$$
$$Y_B = g^{ab} \bmod p$$

Por lo tanto, la llave privada para el usuario A es Y_A y para el usuario B es Y_B, siendo $Y_A = Y_B$. Mientras que X_A y X_B son valores numéricos que se intercambian respectivamente los usuarios A y B para después poder obtener la llave privada. Este método sólo se utiliza para desarrollar e intercambiar una llave privada compartida mediante un canal de comunicación público.

4.7. Firmas Digitales.

Las firmas digitales se generan a partir del cifrado con llave pública haciendo posible certificar la procedencia de un mensaje. Algunos de los sistema de llave pública más empleados para crear firmas digitales son el algoritmo RSA y DSS (Digital Signature Standard) entre otros.

4.7.1. Firma digital con RSA.

Para crear una firma digital con este algoritmo, el emisor cifra la información con su llave privada mientras que el receptor para comprobar la firma debe emplear la llave pública del emisor.
Un usuario cuya llave privada es (d , n) para firmar el mensaje *M* utilizará la función:

$$S = M^d \bmod n$$

donde

S = mensaje firmado

M = mensaje en claro

Cualquier usuario que quiera verificar la procedencia del mensaje, lo podrá hacer con la llave pública del emisor (e, n) aplicando la función:

$$M = S^d \bmod n$$

Por tanto, si un usuario A de llave pública (e_A , n_A) y llave privada (d_A , n_A) quiere enviar un mensaje firmado y cifrado a un usuario B de llave pública (e_B , n_B) y llave privada (d_B , n_B), seguirá el siguiente proceso:
El usuario A firma el mensaje con su llave privada y después lo cifra con la llave pública de B.

$$S = M^{dA} \bmod n_A$$
$$C = S^{eB} \bmod n_B$$

donde

C = mensaje cifrado

El usuario B descifra y verifica la procedencia del mensaje con su llave privada y la llave pública de A.

$$S = C^{dB} \bmod n_B$$
$$M = S^{eA} \bmod n_A$$

4.8. Criptografía de Curvas Elípticas.

Como anteriormente se vio, RSA ha ido incrementando el número de bits de las llaves para ofrecer una mayor seguridad, sin embargo esto ha provocado que el procesamiento de las aplicaciones que usan este método sea más lento. Recientemente, apareció un sistema que está compitiendo con RSA llamado Criptografía de Curvas Elípticas (ECC). La principal ventaja que ofrece ECC con respecto a RSA, es la misma o mayor seguridad con un tamaño más pequeño de bits en las llaves pública y privada, reduciéndose la carga de procesamiento.

Curvas elípticas

Las curvas elípticas son nombradas así porque son descritas por ecuaciones cúbicas, no porque sean elipses. En general, las ecuaciones para curvas elípticas son de la forma

$$y^2+axy+by=x^3+cx^2+dx+e$$

donde a,b,c,d y e son números reales que satisfacen ciertas condiciones. Otro importante elemento en la definición de cualquier curva elíptica es el llamado punto al infinito o punto cero O. Una forma de adición puede ser definida por medio de una curva elíptica, la cual está establecida como: Si tres puntos sobre una curva elíptica caen en una línea recta, la suma es O. De esta definición se pueden establecer las reglas de adición sobre las curvas elípticas de la siguiente manera:

1. O sirve como la identidad aditiva. Entonces O= -O; para cualquier punto P sobre la curva elíptica, P + O =P.
2. Una línea vertical que encuentra la curva elíptica en dos puntos con la misma coordenada x, es decir, $P_1=(x,y)$ y $P_2=(x,-y)$. También encuentra la curva en el punto infinito O. Entonces, $P_1 + P_2 + O = O$, y $P_1 = -P_2$, donde el punto negativo es un punto con la misma coordenada x, pero con la coordenada y negativa (Figura 4.25).
3. Para sumar dos puntos Q y R con diferentes coordenadas en x, se tiene que dibujar una línea recta entre ellos y encontrar un tercer punto de intersección P_1. Hay que notar que $Q + R + P_1 = O$ y $Q + R = -P_1$.
4. Para duplicar el punto Q, se dibuja una línea tangente a la curva para encontrar el otro punto de intersección S. Entonces Q + Q=-S.

Las anteriores reglas de adición satisfacen las leyes de conmutatividad y asociatividad. La multiplicación de un punto P sobre una curva elíptica por un entero positivo k esta definida como la suma de k copias de P.

Curvas elípticas sobre campos finitos

En ECC, se tiene que considerar la forma restringida de las curvas elípticas definida sobre un campo finito. En el caso particular de la criptografía, las curvas elípticas se refieren a grupos elípticos de módulo p, donde p es un

número primo. Este proceso está definido de la siguiente manera: Primero se seleccionan dos números enteros positivos, *a* y *b*, menores a *p* que satisfagan la ecuación

$$4a^3+27b^2 (\text{mod } p) \neq 0$$

Entonces, $E_p(a,b)$ denota el grupo elíptico módulo *p* cuyos elementos (x,y) son pares de números no negativos menores a *p* y que satisfacen la ecuación

$$y^2=x^3+ax+b(mod\ p)$$

junto con el punto al infinito O. Vea siguiente figura.

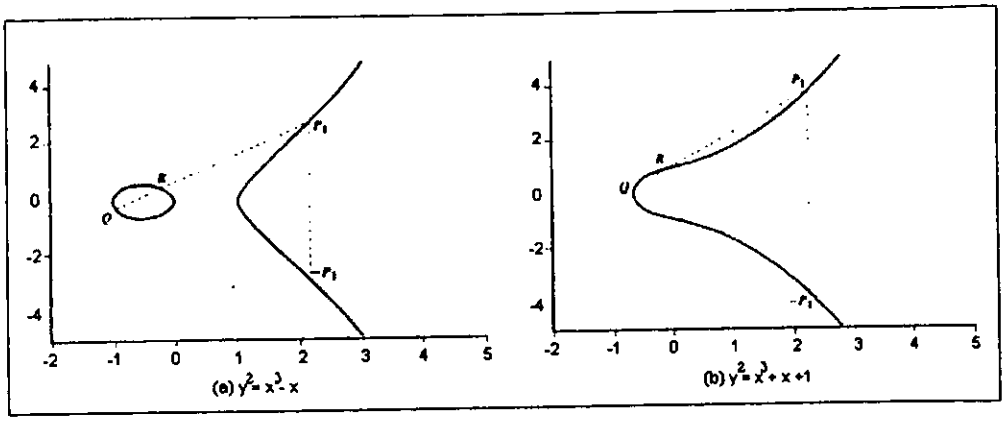


Figura 4.25. Ejemplo de curvas elípticas

Por ejemplo, para p=23 y la curva elíptica $y^2=x^3+x+1$, se tiene que a=b=1 y $4*13+27*12(\text{mod } 23)=8 \neq 0$ que satisface la condición de un grupo elíptico módulo 23. La figura 4.25.b muestra una curva continua con todos los puntos reales que satisfacen la ecuación. Para el grupo elíptico, sólo estamos interesados en los números enteros no negativos en forma de cuadrante (0,0) a (p,p) que satisfacen a la ecuación módulo *p*. La tabla 4.7 lista los puntos que son parte de $E_{23}(1,1)$. Para crear esta lista se realizó los siguientes pasos:

1. Para cada x, tal que $0 \leq x < p$, calcular $x^3+ax+b(mod\ p)$.
2. Para cada resultado del paso anterior, se debe determinar la raíz cuadrada módulo *p*. Si no existe la raíz entonces no existen puntos en $E_p(a,b)$ con el de *x*. En caso contrario existirán dos valores de *y* que satisfacen la operación de la raíz cuadrada (a menos de que el valor sea 0). Los valores de (x,y) obtenidos son los puntos sobre $E_p(a,b)$.

Puntos de la curva elíptica $E_{23}(1,1)$		
(0, 1)	(6, 4)	(12,19)
(0,22)	(6,19)	(13, 7)
(1, 7)	(7,11)	(13,16)
(1,16)	(7,12)	(17, 3)
(3,10)	(9, 7)	(17,29)
(3,13)	(11, 3)	(18, 3)
(4, 0)	(11,20)	(18,20)
(5, 4)	(12, 4)	(19, 5)
(5,19)	(9, 6)	(19,18)

Tabla 4.18. Puntos sobre la curva elíptica $E_{23}(1,1)$.

Las reglas de adición sobre $E_p(a,b)$ corresponden a la técnica geométrica ilustrada en la figura 4.25. Ellas pueden ser establecidas empleando los puntos $P, Q \in E_p(a,b)$:

1. $P + O = P$.
2. Si $O=(x, y)$, entonces $P+(x, -y)=O$. El punto $(x, -y)$ es el negativo de P, denotado como $-P$. En la gráfica de la figura 4.25.b $(x, -y)$ es un punto sobre la curva elíptica y en $E_p(a, b)$. Por ejemplo, en $E_{23}(1, 1)$, para $P=(13,7)$, se tiene $-P=(13, -7)$. Pero $-7 \bmod 23 = 23$, entonces $-P = (13,16)$ está también en $E_{23}(1, 1)$. Ver tabla 4.18.
3. Si $P=(x_1, y_1)$ y $Q(x_2, y_2)$ con $P \neq -Q$, entonces $P + Q = (x_3, y_3)$ determinado por las siguientes reglas:

$$x_3 = \lambda^2 - x_1 - x_2 (mod\ p)$$
$$y_3 = \lambda - (x_1\ x_2) - y_1 (mod\ p)$$

donde

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{para } P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1} & \text{para } P = Q \end{cases}$$

Apliquemos estas ecuaciones en dos ejemplos. Sea $P=(3,10)$ y $Q=(9,7)$, entonces

$$\lambda = \frac{7-10}{9-3} = \frac{-3}{6} = 11 \bmod 23$$
$$x_3 = 11^2 - 3 - 9 = 109 \equiv 17 \bmod 23$$
$$y_3 = 11(3-(-6)) - 10 = 89 \equiv 20 \bmod 23$$

Entonces P+Q = (17, 20). Y para encontrar 2P,

$$\lambda = \frac{3(3)^2 + 1}{2*10} = \frac{5}{20} = 6 \bmod 23$$

$$x_3 = 6^2 - 3 - 3 = 30 \equiv 7 \bmod 23$$

$$y_3 = 6(3 - 7) - 10 = -34 \equiv 12 \bmod 23$$

Por lo que 2P=(7, 12)

Criptografía con curvas elípticas

La operación de adición en ECC es la contraparte de la multiplicación modular en RSA, y la adición multiplicativa es la contraparte de la exponenciación modular. Para formar un sistema criptográfico usando curvas elípticas, necesitamos encontrar dos números primos.

Si consideramos la ecuación Q=kP ,donde Q, P $\in E_p(a,b)$ y k<p, es relativamente fácil calcular Q dado k y P, pero sería difícil si se determina k a partir de Q y P.

Análogía con el intercambio de llaves Diffie-Hellman

El intercambio de llaves usando curvas elípticas puede ser realizado de la siguiente manera. Primero, se toma un número $p \approx 2^{180}$ junto con los parámetros a y b de la curva elíptica par definir al grupo elíptico $E_p(a,b)$. Después se toma un punto generador $G=(x_1,y_1)$ de $E_p(a,b)$. El criterio para seleccionar G es obtener el valor más pequeño de n, para el cual nG= O sea un número primero muy grande.

La llave de intercambio entre usuarios A y B puede ser calculada de la siguiente forma:

1. El usuario A selecciona un número entero n_A menor a n y que corresponde a la llave privada de A. Después, A genera una llave pública $P_A = n_A * G$; la llave pública es un punto en $E_p(a,b)$.
2. De la misma manera, B selecciona una llave privada n_B y calcular una llave pública P_B.
3. Posteriormente, A genera su llave secreta como $k= n_A * P_B$ y B genera su llave secreta calculando $k= n_B * P_B$.

Los dos cálculos en el paso 3 producen el mismo resultado porque

$$n_A * P_B = n_A * (n_B * G) = n_B * (n_A * G) = n_B * P_A$$

Para romper este esquema, un atacante necesitaría calcular k dado G y kG.

Como ejemplo, consideremos p=211; $E_p(0,-4)$ equivalente a la curva $y^2=x^3 - 4$; y G=(2,2) donde nG=O con n=241. La llave privada de A es n_A= 121, entonces la llave pública es P_A= 121(2,2) =(115,48). La llave privada de B es n_B=203, y su llave pública es 203(2,2)=(130,203). La llave secreta compartida es 121(130,203)=203(115,48)=(161,169).

En conclusión, la llave secreta es un par de números. Si esta llave es usada como una llave de sesión para un cifrado convencional, entonces únicamente debe ser generado un número.

Cifrado /Descifrado de Curvas elípticas

La primera tarea en el sistema de cifrar es la de codificar el mensaje *m* para ser enviado como un punto P_m =x-y, el cual será descifrado. Debemos considerar que no se puede simplemente codificar los mensaje como las coordenadas *x* o *y* de un punto, porque no todas las coordenadas están en $E_p(a,b)$.

Así como en el sistema de intercambio de llaves, un sistema de cifrado y descifrado requiere de un punto G y un grupo elíptico $E_p(a,b)$ como parámetros. Cada usuario A selecciona una llave privada n_A y genera una llave pública $P_A=n_A*G$.

Para cifrar y enviar un mensaje P_m a B, A selecciona un número entero aleatorio positivo k y produce el texto cifrado C_m ={ kG, P_m + kP_B }. Observemos que A ha usado la llave pública P_B. Para desencriptar el texto cifrado, B multiplica el primer punto en el par por la llave secreta de B y obtiene el resultado del segundo punto.

$$P_m + kP_B - n_B(kG) = P_m + k(n_A G) - n_B(kG) = P_m$$

A enmascara el mensaje P_m sumándole kP_B. Sólo A conoce el valor de k, aunque P_B es una llave pública, nadie puede remover la máscara kP_B. A también incluye una pista, la cual es suficiente para remover la máscara si se conoce la llave privada n_B.

Un ejemplo del proceso de cifrado es considerar a p=751; $E_p(-1,188)$ cuya curva es $y^2=x^3- x+ 188$; y G=(0,376). Suponer que A desea enviar un mensaje a B que es codificado en el punto elíptico P_m =(562,201). Ahora le corresponde a A seleccionar un número aleatorio, k=386. La llave pública de B es P_B =(201,5). Por lo que se tiene que 386(0,376)=(676,558), y (562,201)+386(0,376)=(676,558) y (562,201)+386(201,5)=(385,328). Entonces A envía el texto cifrado { (676,558), (385,328)}.

Seguridad

La seguridad de ECC depende de que tan difícil es determinar *k* dado *kP* y *P*. Existe una técnica para romper las llaves de ECC conocida como Pollard rho. En la tabla 4.19 muestra una la eficiencia de este método en comparación con la de factorización de un número en dos primos usando la criba del cuerpo de números generalizado.

Tamaño de llave en bits	MIPS-Years	Método de ataque
150	$3.8*10^{10}$	Pollar rho
205	$7.1*10^{18}$	Pollar rho
234	$1.6*10^{28}$	Pollar rho
512	$3*10^{4}$	Criba de campos numéricos generalizado
768	$2*10^{8}$	Criba de campos numéricos generalizado
1024	$3*10^{11}$	Criba de campos numéricos generalizado
1280	$1*10^{14}$	Criba de campos numéricos generalizado
1536	$3*10^{16}$	Criba de campos numéricos generalizado
2048	$3*10^{20}$	Criba de campos numéricos generalizado

Tabla 4.19. Criptoanálisis en curvas elípticas.

4.9. Compendios de mensajes

Existe un esquema de seguridad que no requiere el cifrado del mensaje completo. Este esquema se basa en la idea de una función de dispersión unidireccional, la cual toma una parte arbitraria del texto y a partir de ella calcula una cadena de bits de longitud fija. Esta función de dispersión, llamada compendio de mensaje (MD, Message Digest), tiene tres propiedades importantes:

- Dado el mensaje M, es fácil calcular MD(M).
- Dado MD(M), es posible encontrar M.
- Nadie puede generar dos mensajes que tengan el mismo compendio de mensaje. Para cumplir este criterio, la dispersión debe ser de cuando menos 128 bits de longitud, y de preferencia mayor.

El cálculo de un compendio de mensaje del texto normal es mucho más rápido que el cifrado de ese texto normal con un algoritmo de llave pública, por lo que los compendios de mensaje pueden utilizarse para mandar documentos de texto normal firmados. Si un usuario A, calcula el compendio de mensaje de su texto normal,; luego firma el compendio de mensaje y envía tanto el compendio firmado como el texto normal al usuario B. Si un intruso reemplaza el mensaje M en el camino, el usuario B se dará cuenta de lo ocurrido cuando calcule MD(M).

4.9.1. Algoritmo MD5

El algoritmo MD5 es la quinta de una serie de funciones de dispersión diseñadas por Ron Rivest; opera alterando los bits de una manera complicada, de tal manera que cada bit de salida es afectado por cada bit de entrada.
MD5 procesa el texto de entrada en bloques de 512 bits, divididos a su vez en 16 bloques de 32 bits. La salida del algoritmo es un conjunto de 4 bloques de 32 bits, los cuales concatenados forman un valor de dispersión único de 128 bits. El proceso del algoritmo consiste en los siguientes pasos :

1. *Agregar bits de relleno.* El primer paso consiste en rellenar el mensaje a una longitud de 448 bits (módulo 512), es decir, la longitud del mensaje debe ser 64 bits menor de un múltiplo de 512. El rellenado es siempre agregado, inclusive si el mensaje es de la longitud deseada. Por ejemplo, si el mensaje es de 448 bits, es rellenado por 512 bits para tener una longitud de 960 bits. El relleno de bits consiste en agregar un bit uno al final del mensaje seguido de tantos ceros como sean requeridos (hasta 511). Vea la figura 4.17.

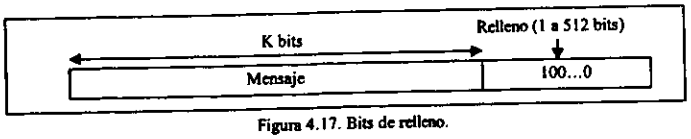


Figura 4.17. Bits de relleno.

2. *Agregar longitud.* Después del campo de relleno del mensaje es agregada la longitud original del mensaje como un entero de 64 bits. Si la longitud original es mayor que 2^{64}, son utilizados los 64 bits de menor orden. Por lo tanto, la longitud del mensaje original es K mod 2^{64}.

   El resultado de los anteriores dos pasos produce un mensaje cuya longitud es un múltiplo de 512 bits. En la siguiente figura, el mensaje expandido es representado por bloques de 512 bits (Y0, Y1, ..., YL-1), por lo que la longitud del mensaje expandido es L*512.

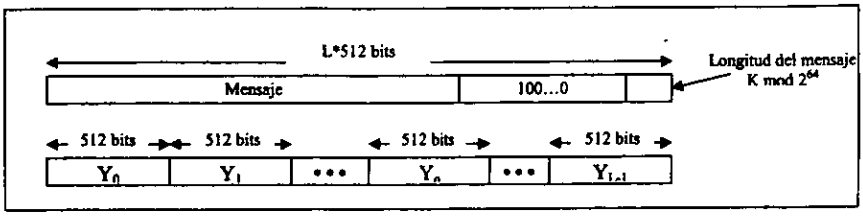


Figura 4.18. Longitud del mensaje expandido

3. *Inicialización del buffer MD.* El último paso de precálculo es la inicialización de un buffer de 128 bits. El buffer se inicializa con la concatenación de cuatro variables de 32 bits llamadas A, B, C y D. El valor de cada variables es: A = 0b01234567, B = 0b89abcdef, C = 0bfedcba98 y D = 0b76543210.

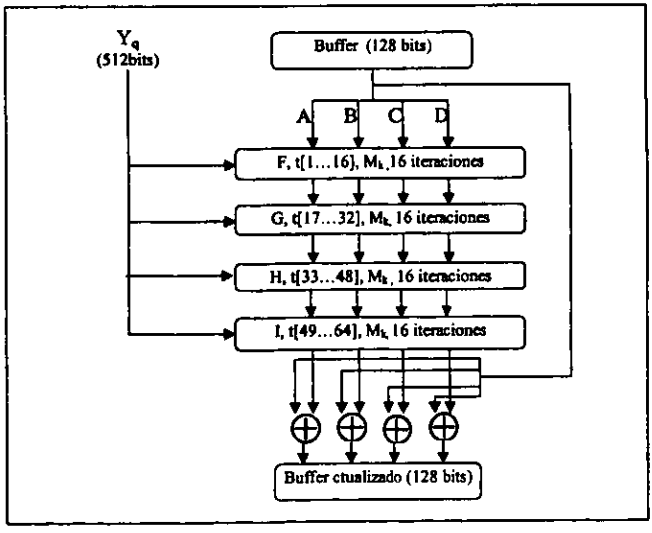


Figura 4.19. Proceso de un bloque de 512 bits.

Posteriormente, las variables son copiadas a otras variables diferentes, a =A, b = B, c = C y d = D.

4. *Procesamiento del mensaje en bloques de 512 bits.* La parte más importante de este algoritmo es la función de compresión y consiste de cuatro rondas. Cada ronda tiene como entrada un bloque Y_q de 512 bits y, el buffer de 128 bits formado por la concatenación de las variables A, B, C y D.

   La salida de la cuarta ronda es sumada con el buffer de entrada de la primera ronda (las variables A, B, C y D), con el fin de actualizar dicho buffer de 128 bits. Además, este valor será la entrada de la primera iteración del siguiente ciclo utilizando el bloque Y_{q+1}.

Compresión en MD5

Cada ronda que se muestra en la figura 4.20, consiste de una secuencia de 16 pasos sobre el buffer ABCD. Cada paso es de la forma:

$$a \leftarrow b + ((a + g(b, c, d) + M_{k(i)} + t_i) <<< s)$$

a, b, c, d = variables obtenidas del buffer ABCD de 128 bits.

g = una de las operaciones F, G, H e I.

<<< s = representa una rotación hacia la izquierda de s bits

$M_{k(i)}$ = k-ésimo bloque de 32 bits obtenido del bloque Y_q.

t_i = el iésimo elemento de la matriz T

\+ = adición modulo 2^{32}

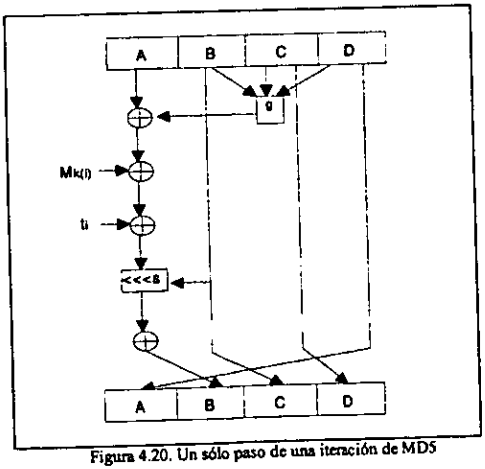


Figura 4.20. Un sólo paso de una iteración de MD5

En cada iteración se usa una de las cuatro operaciones (F, G, H e I), que realizan una función no lineal sobre las variables b, c y d:

Operación	g(b,c,d)
F(b,c,d)	$(b \wedge c) \vee ((\neg b) \wedge d)$
G(b,c,d)	$(b \wedge c) \vee (c \wedge (\neg d))$
H(b,c,d)	$b \oplus c \oplus d$
I(b,c,d)	$c \oplus (b \vee (\neg d))$

Tabla 4.12. Operaciones F, G, H e I; donde $\oplus$ es OR EXCLUSIVA, $\wedge$ es AND, $\vee$ es OR y es $\neg$ NOT.

También se hace uso de una tabla T[t_1... t_{64}] construida a partir de la función seno. Por lo que cada t_i es calculado de la siguiente manera:

$$t_i = 2^{32} * \text{abs}(\sin(i)) \text{ donde i es en radianes}$$

Por otra parte, el orden en que son utilizados los bloques $M_{k(i)}$ es de la siguiente forma:

Ronda	Bloque
1	k (i) = 0,...,15
2	k(i) = (1+5i)mod16
3	k(i) = (5+3i)mod 16
4	k(i) = 7i mod 16

Tabla 4.13. Orden de los bloques $M_{k(i)}$,donde i es el número de paso entre 0 y 64, en las cuatro rondas.

Las cuatro rondas se muestran a continuación:

FF(a, b, c, d, $M_{k(i)}$, s, t_i) denota a = b + ((a + F (b, c, d) + $M_{k(i)}$ + t_i) <<< s)

GG(a, b, c, d, $M_{k(i)}$, s, t_i) denota a = b + ((a + G (b, c, d) + $M_{k(i)}$ + t_i) <<< s)

HH(a, b, c, d, $M_{k(i)}$, s, t_i) denota a = b + ((a + H (b, c, d) + $M_{k(i)}$ + t_i) <<< s)

II(a, b, c, d, $M_{k(i)}$, s, t_i) denota a = b + ((a + I (b, c, d) + $M_{k(i)}$ + t_i) <<< s)

A continuación se muestran en forma explícita los pasos de cada ronda de MD5:

Ronda 1	Ronda2	Ronda3	Ronda4
FF(a,b,c,d,M_0, 7,0xd76aa478)	GG(a,b,c,d,M_1, 5,0xd76aa478)	HH(a,b,c,d,M_5, 4,0xfffa3942)	II(a,b,c,d,M_0, 6,0xf4292244)
FF(a,b,c,d,M_1,12,0xe8c7b756)	GG(a,b,c,d,M_6, 9,0xf61e2562)	HH(a,b,c,d,M_8,11,0x8771f681)	II(a,b,c,d,M_7,10,0x432aff97)
FF(a,b,c,d,M_2,17,0x242070db)	GG(a,b,c,d,M_{11},14,0xc040b340)	HH(a,b,c,d,M_{11},16,0x6d9d6122)	II(a,b,c,d,M_{14},15,0xab9423a7)
FF(a,b,c,d,M_3,22,0xc1bdceee)	GG(a,b,c,d,M_0, 20,0x265e5a51)	HH(a,b,c,d,M_{14},23,0xfde5380c)	II(a,b,c,d,M_5,21,0xfc93a039)
FF(a,b,c,d,M_4, 7,0xf57c0faf)	GG(a,b,c,d,M_5, 5,0xe9b6c7aa)	HH(a,b,c,d,M_1, 4,0x4bdecfa9)	II(a,b,c,d,M_{12}, 6,0x655b59c3)
FF(a,b,c,d,M_5,12,0x4787c62a)	GG(a,b,c,d,M_{10}, 9,0xd62f105d)	HH(a,b,c,d,M_4,11,0x4bdecfa9)	II(a,b,c,d,M_3,10,0x8f0ccc92)
FF(a,b,c,d,M_6,17,0xa8304613)	GG(a,b,c,d,M_{15},14,0x02441453)	HH(a,b,c,d,M_7,16,0xf6bb4b60)	II(a,b,c,d,M_{10},15,0xffeff47d)
FF(a,b,c,d,M_7,22,0xfd69501)	GG(a,b,c,d,M_4, 20,0xd8a1e681)	HH(a,b,c,d,M_{10},23,0xbebfbc70)	II(a,b,c,d,M_1,21,0x85845dd1)
FF(a,b,c,d,M_8, 7,0x698098d8)	GG(a,b,c,d,M_9, 5,0xe7d3fbc8)	HH(a,b,c,d,M_{13}, 4,0x289b7ec6)	II(a,b,c,d,M_8, 6,0x6fa87e4f)
FF(a,b,c,d,M_9,12,0x8b44f7af)	GG(a,b,c,d,M_{14}, 9,0x21e1cde6)	HH(a,b,c,d,M_6,11,0xcaa127fa)	II(a,b,c,d,M_{15},10,0xfe2ce6e0)
FF(a,b,c,d,M_{10},17,0xffff5bb1)	GG(a,b,c,d,M_3,14,0xf4d50d87)	HH(a,b,c,d,M_3,16,0xd4ef3085)	II(a,b,c,d,M_6,15,0xa3014314)
FF(a,b,c,d,M_{11},22,0x89cd7be)	GG(a,b,c,d,M_8,20,0x455a14de)	HH(a,b,c,d,M_6,23,0x04881d05)	II(a,b,c,d,M_{13},21,0x4e0811a1)
FF(a,b,c,d,M_{12}, 7,0x6b901122)	GG(a,b,c,d,M_{13}, 5,0xa9e3e905)	HH(a,b,c,d,M_9, 4,0xd9d4d039)	II(a,b,c,d,M_4, 6,0xf7537e82)
FF(a,b,c,d,M_{13},12,0xfd987193)	GG(a,b,c,d,M_2, 9,0xfcefa3f8)	HH(a,b,c,d,M_{12},11,0xe6db99e5)	II(a,b,c,d,M_{11},10,0xdb3af235)
FF(a,b,c,d,M_{14},17,0xa679438e)	GG(a,b,c,d,M_7,14,0x676f02d9)	HH(a,b,c,d,M_{15},16,0x1fa27cf8)	II(a,b,c,d,M_2,15,0x2ad7d2bb)
FF(a,b,c,d,M_{15},22,0x49b40821)	GG(a,b,c,d,M_{12},20,0x8d2a4c8a)	HH(a,b,c,d,M_2,23,0xc4ac5665)	II(a,b,c,d,M_9,21,0xeb86d391)

Tabla 4.14. Pasos de cada ronda de MD5.

El compendio del mensaje se obtiene al ser procesado el último bloque de 512 bits Y_{L-1}, siendo de 128 bits. Vea la siguiente figura.

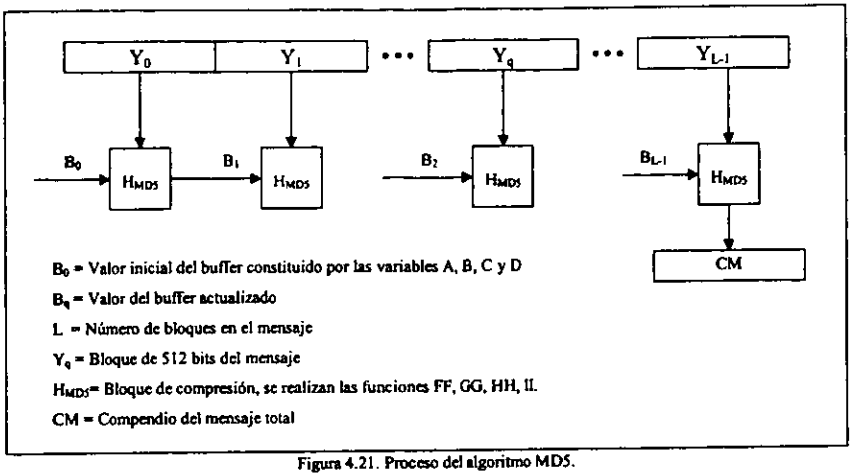


Figura 4.21. Proceso del algoritmo MD5.

Seguridad

El algoritmo de MD5 tiene la propiedad de que cada bit del código de dispersión es una función de cada bit de entrada. Las funciones básicas F, G, H, I, producen una mezcla efectiva de los bits de entrada y parecería un buen código de seguridad. Sin embargo, han existido ataques contra MD5, uno muy serio fue desarrollado por Dobbertin. Su técnica consiste en generar una colisión en la función de compresión MD5, el ataque trabaja en una operación de MD5 sobre un bloque de 512 bits de entrada encontrando otro bloque que produce la misma salida de 128 bits. A pesar de lo anterior, no se ha encontrado una forma de generalizar el ataque en todo el mensaje usando el valor inicial del buffer.

4.9.2. Algoritmo Seguro de Dispersión (SHA).

El Algoritmo Seguro de Dispersión (SHA, Secure Hash Algorithm) fue desarrollado por el NIST (National Institute of Stardard) en 1993. Al igual que el MD5, SHA procesa datos de entrada en bloques de 512, pero la diferencia es que genera un compendio de mensaje de 160 bits. El proceso del algoritmo consiste en los siguientes pasos:

1. *Agregar bits de relleno.* El primer paso consiste en rellenar el mensaje a una longitud de 448 bits (módulo 512), es decir, la longitud del mensaje debe ser 64 bits menor de un múltiplo de 512. El rellenado es siempre agregado, inclusive si el mensaje es de la longitud deseada. El número de bits de relleno está en el rango de 1 a 512. El relleno de bits consiste en agregar un bit uno al final del mensaje seguido de ceros.

2. *Agregar longitud.* Después del campo de relleno del mensaje es agregada la longitud original del mensaje en 64 bits.
3. *Inicialización del buffer.* Se utiliza un buffer de 160 bits para generar el compendio de mensaje. El buffer se inicializa con la concatenación de cinco variables de 32 bits llamadas A, B, C, D y E. El valor de cada variables es: A = 0x67452301, B = 0xefcdadab89, C= 0x98badcfe , D = 0x10325476 y E = 0xc3d2e1f0. Posteriormente, las variables son copiados a otras variables diferentes, a =A, b = B, c = C , d = D y e=E.
4. *Procesamiento del mensaje en bloques de 512 bits.* El algoritmo empieza cuando se procesa un bloque Y_q de 512 bits del mensaje. La parte más importante de este algoritmo es la función de compresión y consiste de cuatro rondas de 20 operaciones cada una. Cada operación realiza una función no linear sobre tres variables: a, b, c, d y e. El conjunto de funciones no lineales de SHA son:

Paso	Operación	g(b,c,d)
0<=t<=19	f_1= f(t, b ,c ,d)	$(b \wedge c) \vee ((\neg b) \wedge d)$
20<=t<=39	f_2= f(t, b ,c ,d)	$b \oplus c \oplus d$
40<=t<=59	f_3= f(t, b ,c ,d)	$b \wedge c) \vee (b \wedge d) \vee (c \wedge d)$
60<=t<=79	f_4= f(t, b ,c ,d)	$b \oplus c \oplus d$

Tabla 4.15. Funciones no lineales de SHA, donde $\oplus$ es OR EXCLUSIVA, $\wedge$ es AND, $\vee$ es OR y $\neg$ es NOT.

Cada ronda hace uso de una constante aditiva K, donde 0 <= t <= 79 indica uno de los 80 pasos de las cuatro rondas. Las cuatro constantes usadas en el algoritmo son: K_t=0x5a827999, desde t=0 a 19; K_t=0x6ed9eba1, desde t=20 a 39; K_t=0x8f1bbcdc, desde t=40 a 59 y K_t=0xca62c1d6, desde t=60 a 79.

El buffer inicial (concatenación de los valores de A, B, C, D y E), es utilizado, al igual que el bloque Y_0 de 512 bits del mensaje, como entrada en la primera ronda (Figura 4.22). Esta entrada es procesada utilizando las funciones f_t y la constante K_t, dando como resultado una salida que será la entrada de siguiente ronda. Esta operación se repite en las siguientes dos rondas. La salida de la cuarta ronda es sumada con la entrada de la primer ronda y el resultado es utilizado en el proceso del siguiente bloque Y_{q+1}, de esta manera se actualiza el buffer (A, B, C, D y E). El proceso anterior se repite para los L bloques de 512 bits que forman el mensaje total, obteniendo un compendio de mensaje de 160 bits del resultado de la última ronda del bloque Y_{L-1}.

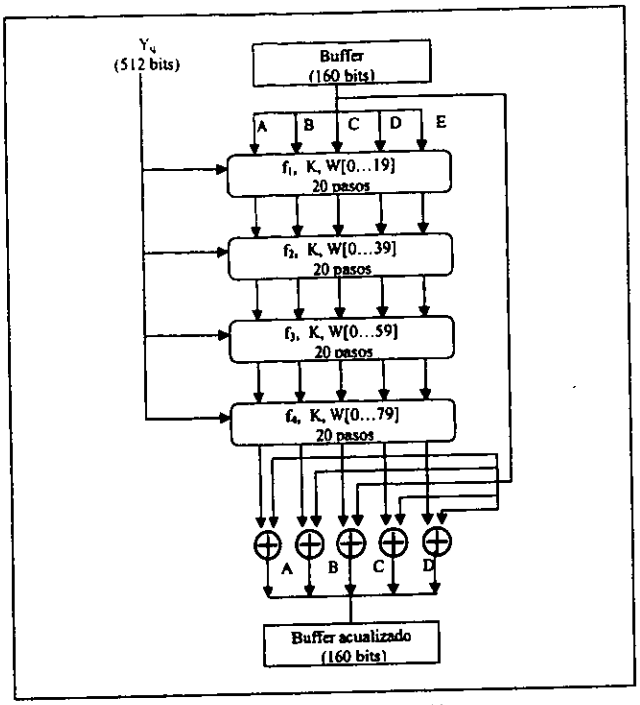


Figura 4.22. Procesamiento de un bloque de 512 bits.

Función de compresión

Cada ronda es de la siguiente forma:

$$A,B,C,D,E \leftarrow (E + f(t,B,C,D) + S^5(A) + W_t + K_t), A, S^{30}(B), C, D$$

donde

A,B,C,D,E =las cuatro palabras del buffer

t= el número de pasos; 0<=t<79

f(t, B, C, D) = función no lineal para el paso t

S^k = rotación circular a la izquierda de 32 bits de k bits

W_t = una palabra de 32 bits derivada de la entrada de bloque de 512 bits

K_t = es una constante aditiva. Cuatro valores distintos son usados.

+ = Adición módulo 2^{32}

Este cálculo se observa en la siguiente figura.

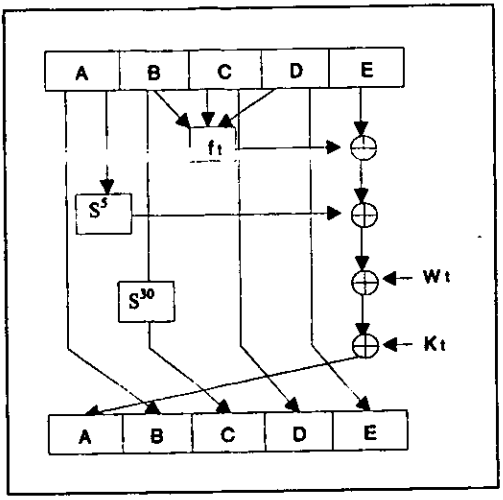


Figura 4.23. Un paso del algoritmo SHA.

Cada bloque es transformado de 16 palabras de 32 bits (Y_0 a Y_{15}), a 80 palabras de 32 bits (W_0 a W_{79}) usando el siguiente algoritmo:

$$Wt = Mt \text{ desde } t = 0 \text{ a } 15$$

$$Wt = S^1 (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}), \text{ desde } t = 16 \text{ a } 79$$

Si t es el número de operación (desde 1 a 80), Wt representa el bloque enésimo del mensaje expandido, y S^1 representa un corrimiento circular a la izquierda de 1 bit. Ver figura 4.24.

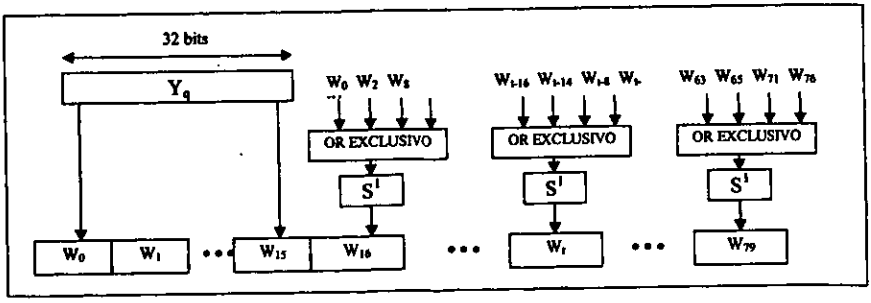


Figura 4.24. Creación de bloques de 80 palabras de 32 bits.

Seguridad

En este algoritmo de dispersión no se usa ninguna tabla de números aleatorios (ni valores senoidales), pero por cada bloque se calculan 80 rondas generando una mezcla exhaustiva. Puesto que el código de dispersión de SHA es 32 bits más largo que el de MD5, el primero es 2^{32} más seguro; sin embargo, es más lento. Todavía no se sabe sí este algoritmo ha sido vulnerable a un ataque.

El SHA es un estándar del gobierno, y lo usan compañías que tienen que usarlo porque el gobierno les dice que lo hagan, o aquellas que quieren seguridad extra.

4.10. SSL.

SSL (Secure Socket Layer, Capa del Extremo Seguro) es una especificación propietaria de Netscape puesta en dominio público para la definición de canales seguros sobre TCP, el protocolo de transporte punto a punto de Internet. El objetivo inicial era la realización de conexiones seguras a servidores Web que permitiera, por ejemplo, enviar números de tarjetas de crédito a través de un formulario.

La ventaja de SSL es que es un protocolo de aplicación independiente del protocolo de comunicaciones (HTTP, FTP, TELNET, etc.), encargado de negociar el algoritmo de encriptación y la llave de la sesión en forma transparente. El SSL puede negociar el algoritmo de encriptación y autentificar el servidor antes de que el protocolo de transmisión reciba o envíe datos. Después, todos los datos son transmitidos en forma encriptada para asegurar la privacidad.

4.10.1. Arquitectura del SSL.

El protocolo SSL se compone de dos capas conformadas por: El Protocolo de Registro (Record Protocol) SLL, que ocupa la capa inferior, encargado de encapsular los protocolos de nivel más alto. La capa superior está constituida por el Protocolo de Negociación (Handshake Protocol), el Protocolo de Especificación de Cambio de Cifrado (Change Cipher Spec Protocol) y el Protocolo de Alerta (Alert Protocol). Observe la figura 4.26.

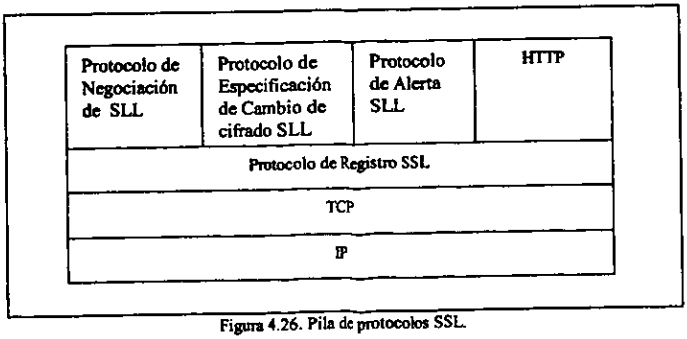


Figura 4.26. Pila de protocolos SSL.

Dos importantes conceptos que se manejan en SSL son la sesión y la conexión, definido en la especificación de SSL de la siguiente manera:

- **Conexión**. Una conexión es un transporte que proporciona el tipo adecuado de servicio. En SSL las conexiones son relaciones punto a punto. Las conexiones son transitorias y cada una de ellas esta asociada a una sesión.
- **Sesión**. Una sesión SSL es una asociación entre un cliente y un servidor creada por el Protocolo de Negociación. Las sesiones definen un conjunto de parámetros de seguridad criptográfica, que pueden ser compartidos entre conexiones múltiples. El propósito de estas sesiones es que en cada conexión ya no se involucren nuevos parámetros de seguridad, evitándose así una negociación costosa.

Existe actualmente un número de estados asociados a cada sesión. Una vez que la sesión se ha establecido, existe un estado operativo actual para leer y escribir. En el Protocolo de Negociación, los estados de lectura y escritura en espera son creados. Después de una exitosa conclusión del Protocolo de Negociación, los estados de espera llegan a ser estados actuales. Un estado de sesión está definido por los siguientes parámetros (definiciones de la especificación de SSL):

1. *Identificación de sesión.* Es una secuencia arbitraria de bytes seleccionada por el servidor para identificar si el estado de la sesión es activo o reanudable.
2. *Certificación de extremo.* Es un certificado *X509.v3* del extremo. Este elemento del estado puede ser nulo.
3. *Método de compresión.* El algoritmo usado para comprimir datos antes de la encriptación.
4. *Especificación de cifrado.* Especifica el algoritmo de cifrado (por ejemplo, nulo, DES, etc.) y el algoritmo de dispersión (como MD5 ó SHA) usado para calcular el código de autentificación del mensaje (MAC).
5. *Secreto maestro.* Secreto de 48 bytes compartido entre el cliente y el servidor.
6. *Reanudar.* Una bandera que indica si la sesión puede ser usada para iniciar nuevas conexiones.

Un estado de conexión está definido por los siguiente parámetros:

1. *Servidor y cliente aleatorio.* Secuencias de bytes establecidas por el servidor y el cliente en cada conexión.
2. *Secreto MAC de escritura del servidor.* Llave secreta usada en operaciones MAC sobre los datos enviados por el servidor.
3. *Secreto MAC de escritura del cliente.* Llave secreta usada en operaciones MAC sobre los datos enviados por el cliente.
4. *Llave de escritura del servidor.* Llave de encriptamiento convencional utilizada para cifrar datos por el servidor y descifrarlos por del cliente.
5. *Llave de escritura del cliente.* Llave de encriptamiento convencional para cifrar datos por el cliente y descifrarlos por el servidor.

6. *Vectores de iniciación.* Cuando se emplea un algoritmo de cifrado de bloques, la iniciación del vector (IV) es mantenida en cada llave. Este campo es iniciado por el Protocolo de Negociación SSL; a partir de entonces el último bloque de texto cifrado de cada registro es conservado para usarlo como el vector IV en el siguiente registro.
7. *Número de secuencia.* Cada sesión mantiene números de secuencias separadas para los mensajes recibidos y transmitidos en cada conexión. Cuando un extremo de la comunicación envía o recibe un mensaje de especificación de cambio de cifrado, el número de secuencia es establecido en cero. Los número de secuencia no deben exceder de 2^{64}-1.

4.10.2. Protocolo de Registro.

El Protocolo de Registro de SSL proporciona dos servicios para conexiones SSL:

- *Confidencialidad.* El Protocolo de Negociación define una llave secreta compartida que es usada para el encriptamiento convencional de cargas útiles de SSL.
- *Integridad de mensajes.* El protocolo de Negociación también define una llave secreta compartida que es usada para formar un código de autentificación de mensaje (MAC).

El Protocolo de Registro SSL se encarga de tomar el mensaje de la capa de aplicación, los datos los fragmenta en bloques manejables, opcionalmente los comprime, después aplica MAC, lo encripta, le agrega una cabecera y transmite la unidad resultante en un segmento TCP. Los datos recibidos son desencriptados, verificados, descomprimidos, vueltos a ensamblar y liberados a los usuarios de alto nivel. La siguiente figura indica las operaciones del Protocolo de Registro SSL.

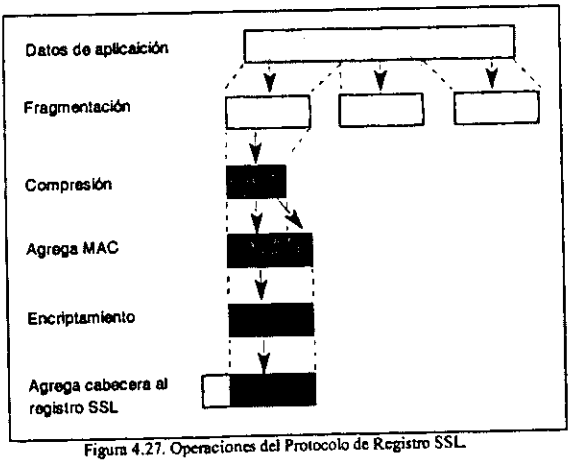


Figura 4.27. Operaciones del Protocolo de Registro SSL.

El primer paso es la fragmentación. Cada mensaje de la capa superior es fragmentado en bloques de 2^{14} bytes (16384 bytes) o menos. El paso de compresión es opcional, si este es llevado a cabo no debe haber pérdidas y no debe incrementarse el contenido del mensaje por más de 1024 bytes. En SLL v3 ningún algoritmo de compresión está especificado, por defecto el algoritmo de compresión es nulo.

El siguiente paso de procesamiento es calcular el código de autentificación de mensaje sobre los datos comprimidos. Para este propósito es usada una llave secreta compartida. El cálculo está definido como sigue:

```
hash(secreto_escritura_MAC || bloque_2 ||
      hash(secreto_escritura_MAC || bloque_1|| num_sec || compresiónSSL.tipo ||
           compresiónSLL.longitud || compresiónSSL.fragmento))
```

donde

||	= concatenación
secreto_escritura_MAC	= llave secreta compartida
hash	= algoritmo de dispersión criptográfico MD5 ó SHA.
bloque_1	= byte 0x36 repetido 48 veces (384 bits) para MD5 y 40 veces (320 bits) para SHA.
bloque_2	= byte 0x5C repetido 48 veces para MD5 y 40 veces para SHA.
num_sec	= número de secuencia para este mensaje.
compresiónSSL.tipo	= tipo de protocolo de capa superior usado para procesar este fragmento.
compresiónSSL.longitud	= longitud del fragmento comprimido.
CompresiónSSL.fragmento	= el fragmento comprimido (si la compresión no es usada, el fragmento es el texto natural).

Posteriormente, el mensaje comprimido más el MAC son cifrados usando el encriptamiento simétrico. El encriptamiento no debe incrementar la longitud del contenido por más de 1024 bytes, entonces la longitud total no debe exceder de 2^{14}+ 2048. Los algoritmos de cifrado de la tabla 4.20 están permitidos.

Algoritmo	Tipo de cifrado	Tamaño de la llave
IDEA	Bloque	128
RC2-40	Bloque	40
DES-40	Bloque	40
DES	Bloque	54
3DES	Bloque	168
Fortezza	Bloque	80
RC4-40	Flujo de datos	40
RC4-128	Flujo de datos	128

Tabla 4.20. Lista de algoritmos de encriptamiento en SSL

Fortezza es particularmente usado en un esquema de encriptamiento en tarjetas inteligentes.

Para el cifrado de flujo de datos, el mensaje comprimido más el MAC son encriptados (Figura 4.28). El MAC es calculado antes de que el cifrado ocurra, después es encriptado junto con el texto natural o el texto comprimido.

Para el encriptamiento de bloque, la carga útil puede ser agregada después del MAC y antes del encriptamiento. La cantidad total de carga útil es la cantidad más pequeña tal que el tamaño total de los datos a ser encriptados es un multiplo de la longitud del bloque cifrado. Por ejemplo, un texto natural (o comprimido) de 58 bytes con un MAC de 20 bytes (con SHA), es encriptado usando un bloque de longitud de 8 bytes. Con el byte de longitud de carga, esto produce un total de 79 bytes. Para que sea múltiplo de 8 es agregado un byte de carga.

El paso final del proceso del Protocolo de Registro SSL es agregar una cabecera con los campos mostrados en la figura 4.28 y descritos a continuación:

1. *Tipo de contenido* (8 bits). El protocolo de la capa superior usado para procesar el encapsulado del fragmento.
2. *Versión mayor* (8 bits). Indica la versión más reciente de SSL en uso. Para SSLv3, el valor es 3.
3. *Versión menor* (8 bits). Indica la versión menor en uso. Para SSLv3, el valor es 0.
4. *Longitud comprimida* (16 bits). La longitud en bytes del fragmento de texto natural (o comprimido). El valor máximo es 2^{14}+2048.

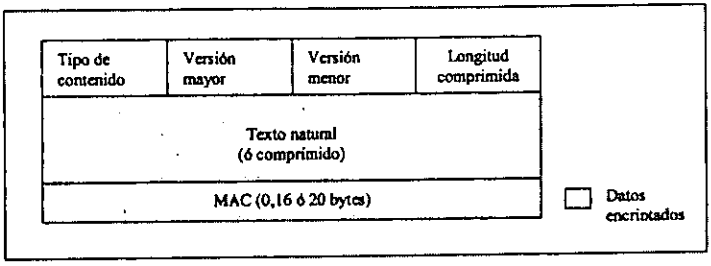


Figura 4. 28. Formato del registro SSL

4.10.3. Protocolo de Especificación de Cambio de Cifrado.

Este protocolo consiste en un solo mensaje, el cual contienen un solo bit con el valor 1 (Figura 4.29.a). El único propósito de este mensaje es crear el estado de espera para ser copiado en el actual estado y así actualizar el conjunto de algoritmos de cifrado a ser usados en la conexión.

4.10.4. Protocolo de Alerta.

El Protocolo de Alerta es usado para transportar alertas relacionadas con SSL al extremo de la comunicación. Como se observa en la figura 4.29.b, el primer byte toma el valor de advertencia (1) o fatal (2) para transportar la severidad del mensaje. Si el nivel es fatal, SSL inmediatamente termina la conexión. Otras conexiones sobre la misma sesión pueden continuar, pero no nuevas conexiones sobre esta sesión pueden establecerse. El segundo byte contiene un código que indica la especificación de alerta. Primero se listan aquellas alertas que siempre son fatales:

1. *mensaje_inesperado*. Un mensaje inapropiado fue recibido.
2. *registro_incorrecto_mac*. Fue recibido en forma incorrecta el código de autentificación de mensaje.
3. *falla_descompresión*. La función de descompresión recibe una entrada impropia, por lo que será incapaz de descomprimir.
4. *falla_negociación*. El emisor es incapaz de negociar un conjunto aceptable de parámetros de seguridad.

5. *parámetro_ilegal.* Un campo en el mensaje de negociación está fuera de rango o es inconsistente con otros campos.
6. *notificar_cerrado.* Notifica que el emisor no enviará más mensajes sobre esa conexión.
7. *no_certificado.* Puede ser enviado en respuesta a una petición de certificado si no esta disponible un certificado apropiado.
8. *certificado_inadecuado.* Un certificado recibido fue corrompido, por ejemplo una firma no es válida.
9. *no_soporta_certificado.* El tipo de certificado recibido no es soportado.
10. *certificado_revocado.* Un certificado ha sido revocado por su firmante.
11. *certificado_expirado.* Un certificado ha expirado.
12. *certificado_no_conocido.* Algunos otros asuntos no especificados surgieron en el proceso del certificado.

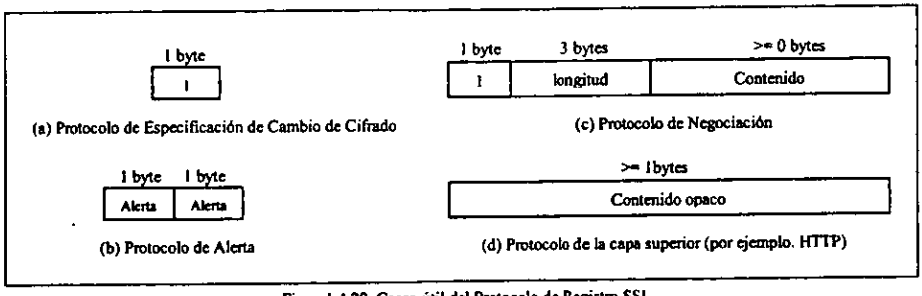


Figura1 4.29. Carga útil del Protocolo de Registro SSL.

4.10.5. Protocolo de Negociación.

La parte más compleja de SSL es el Protocolo de Negociación. Este protocolo permite tanto al servidor y como al cliente autentificarse entre ellos y negociar el cifrado, el algoritmo MAC y las llaves de cifrado usadas para proteger los datos enviados en un registro SSL. El Protocolo de Negociación es usado antes de que cualquier dato de aplicación sea transmitido.

El Protocolo de Negociación consiste de una serie de mensajes de intercambio por el cliente y el servidor. Todos estos tienen un formato mostrado en la figura 4.29(c). Cada mensaje tiene tres campos:

- *Tipo* (1 byte). Indica uno de los 10 mensajes de la tabla 4.22.
- *Longitud* (3 bytes). La longitud de mensajes en bytes.
- *Contenido* (>=1byte). Los parámetros asociados con el mensaje; éstos son listados en la tabla 4.22.

Tipo de mensaje	Parámetros
solicitud_hola	Nulo
cliente_hola	versión, ramdom, ID sesión, método de compresión.
servidor_hola	versión, random, ID sesión, método de compresión.
Certificado	cadena de certificados X.509v3.
servidor_intercambio_llave	parámetros, firma
solictud_certificado	tipo, autoridades
servidor_hecho	nulo
verificar_certificado	firma
cliente_intercambio_llave	parámetros, firma.
Terminada	valor disperso

Tabla 4.21. Tipo de mensajes del Protocolo de Negociación SSL.

En la figura 4.30 se muestra el intercambio inicial para establecer una conexión lógica entre cliente y servidor. El intercambio puede ser visto en cuatro fases.

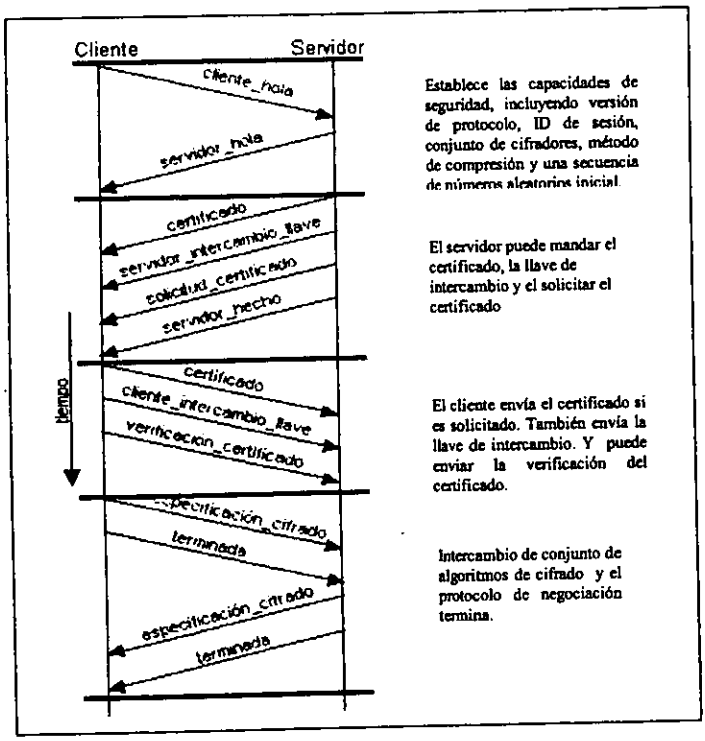


Figura 4.30. Acción del Protocolo de Negociación.

Fase 1. Establece las capacidades de seguridad

Esta fase es usada para iniciar una conexión lógica y establecer las capacidades de seguridad que serán asociadas con ella. El intercambio es iniciado por cliente, el cual envía el mensaje *cliente_hola* con los siguientes parámetros:

1. *Versión.* La más reciente versión de SSL entendida por el cliente.
2. *Random.* El cliente genera una estructura aleatoria que consiste en una estampilla de tiempo de 32 bits y 28 bytes creados por un generador de números aleatorios. Estos valores son usados durante el intercambio de llave para prevenir ataques.
3. *ID sesión.* Es un identificador de sesión. Un valor diferente a cero indica que el cliente desea actualizar los parámetros de una conexión existente o crear una nueva conexión sobre esa sesión. Un cero indica que el cliente desea establecer una nueva conexión sobre una nueva sesión.
4. *Conjunto de algoritmos de cifrado.* Esta es una lista que contiene las combinaciones de algoritmos criptográficos soportados por el cliente, en orden decreciente según la preferencia. Cada elemento de la lista define un algoritmo de intercambio de llave y la especificación del algoritmo de cifrado.
5. *Método de compresión.* Es una lista de los métodos de compresión que el cliente soporta.

Después de enviar el mensaje *cliente_hola*, el cliente espera el mensaje *servidor_hola*, el cual contiene los mismos parámetros del primero. Para el mensaje *servidor_hola* las siguientes convenciones se aplican. El campo *Versión* contiene la más baja de las versiones sugerida por el cliente y la más alta soportada por el servidor. El campo *Random* es generado por el servidor y es independiente del campo *Random* del cliente. Si el campo *ID sesión* del cliente fuera cero, el mismo valor es usado por el servidor; de otra forma el campo ID sesión del servidor contiene el valor de una nueva sesión. El campo de *Conjunto de algoritmos de cifrado* contiene un solo conjunto de cifradores seleccionados por el servidor que fueron propuestos por el cliente. El campo *Método de compresión* contiene el método de compresión seleccionado por el servidor que fueron propuestos por el cliente.
El primer elemento de los parámetros del conjunto de algoritmos de cifrado es el método de intercambio de llave.
Los siguientes cinco métodos de intercambio de llave son soportados:

- RSA. La llave secreta es encriptada con la llave pública RSA del receptor. Un certificado de llave pública para la llave del receptor debe estar disponible.
- Diffie-Hellman fijo. Este es un intercambio de llave Diffie-Hellman, en el cual el certificado del servidor contiene los parámetros públicos Diffie-Helman firmados por la autoridad certificadora (CA). Esto es el certificado de llave pública contiene los parámetros de llave pública Diffie-Hellman. El cliente proporciona sus parámetros de llave pública Diffie-Hellman, inclusive en un certificado sí la autentificación del cliente es requerida o, en un mensaje de intercambio de llave.
- Diffie-Hellman efímero. Esta técnica es usada para crear llaves secretas efímeras (temporales). En este caso, las llaves públicas Diffie-Hellman son intercambiadas, firmadas usando la llave privada RSA o la llave DSS del

emisor. El receptor puede usar la llave pública correspondiente para verificar la firma. Los certificados son usados para autentificar las llaves públicas. Esto parece ser el método más seguro de las tres opciones Diffie-Hellman porque la autentificación de llave es en un tiempo.

- Diffie-Hellman anónimo. El algoritmo Diffie-Hellman es usada sin autentificación. Esto es, cada lado envía entre sí sus parámteros públicos Diffie-Helmman. Este aspecto es vulnerable a ataques donde un husmeador puede llevar a cabo un Diffie-Hellman anónimo en ambos extremos.
- Fortezza. La técnica definida por el esquema de Fortezza.

En el método de intercambio de llave, la especificación del algoritmo de cifrado incluye los siguiente campos:

1. *Algoritmo de cifrado.* Cualquiera de los algoritmos mencionados: RC4, RC2, DES, 3DES, DES40, IDEA, Fortezza.
2. *Algortimo MAC.* MD5 o SHA-1.
3. *Tipo de algoritmo de cifra.* Bloque o flujo de datos.
4. *Exportable.* Verdadero o falso.
5. *Tamaño del compendio.* 0, 16 (MD5), ó 20(SHA-1) bytes.
6. *Material de llave.* Una secuencia de bytes que el contenido de datos usó en generar las llaves de escritura.
7. *Tamaño del vector IV.* El tamaño del valor de iniciación para la Cadena de Bloque de Cifrado (CBC) .

Fase 2. Autentificación del servidor e intercambio de llave

El servidor empieza esta fase enviando su certificado si necesita ser autentificado; el mensaje contiene uno o una cadena de certificados X.509. El certificado de mensaje es requerido para cualquier método de intercambio de llaves excepto para el Diffie-Hellman anónimo. Si es usado Diffie-Helmman fijo, el mensaje certificado funciona como mensaje de intercambio de llave del servidor porque contiene parámetros Diffie-Hellman públicos del servidor. Después el mensaje *servidor _ intercambio_ llave* puede ser enviada si es requerido, en caso contrario existen dos opciones: El servidor ha enviado un certificado con los parámetros Diffie-Hellman fijo, o el intercambio de llave usado es usado. El mensaje *servidor _ intercambio_ llave* es necesitado por los siguientes cuatro métodos:

- Diffie-Hellman anónimo. El contenido del mensaje consiste de dos valores globales Diffie-Hellman (un número primo y la raíz primitiva del número).
- Diffie-Hellman fijo. El contenido del mensaje incluye tres parámetros Diffie-Hellman proporcionados por Diffie-Hellman anónimo, más una firma de aquellos parámetros.
- Intercambio de llave RSA, en el cual el servidor usa la llave con una sola firma. El cliente no puede enviar una llave secreta encriptada con la llave pública del servidor. En lugar de ello, el servidor debe crear un par de llaves pública /privada RSA temporales. El contenido del mensaje incluye dos parámetros de la llave pública RSA temporal más una firma de aquellos parámetros.
- Fortezza

Una firma es creada tomando el compendio de un mensaje y encriptandolo con la llave pública del emisor. En este caso el compendio está definido como :

Hash(ClienteHola.aleatorio || ServidorHola.aleatorio || ServidorParámetros)

Entonces el compendio no sólo cubre los parámetros Diffie-Hellman o RSA, también los dos actuales del mensaje inicial hola. Esto evita réplicas de ataques y palabras tergiversadas. En el caso de una llave DDS, el compendio es realizado usando el algoritmo SHA-1. En el caso de una firma RSA, el compendio es calculado mediante MD5 y SHA-1, y la concatenación de los dos son encriptados con la llave pública del servidor.
Después, un servidor no anónimo (servidor que no usa Diffie-Hellman anónimo) puede solicitar un certificado del cliente. El mensaje *petición_certificado* incluye dos parámetros: *tipo_certificado* y *autoridades_*certificado.
El *tipo_certificado* indica el algoritmo de llave pública y puede usar:

- RSA, únicamente firma.
- DSS, únicamente firma.
- RSA para Diffie-Hellman fijo; en este caso la firma es usada únicamente para la autentificación, enviando un certificado firmado con RSA.
- DSS para Diffie-Hellman fijo, usado únicamente para la autentificación.
- RSA para Diffie-Hellman fijo.
- DSS para diffie-Hellman fijo.
- Fortezza.

El segundo parámetro en el mensaje *petición_ certificado* es una lista de los nombres distinguidos de autoridades certificadas aceptables.
El mensaje final en la fase 2, y uno que siempre es requerido, es el mensaje *servidor_hecho*, el cual es enviado por el servidor para indicar el fin del hola del servidor y mensajes asociados. Después de enviar este mensaje, el servidor esperará por la respuesta del cliente. Este mensaje no tiene parámetros.

Fase 3. Autentificación del cliente y la llave de intercambio

Sobre el recibo del mensaje *servidor_hecho*, el cliente deberá verificar que el servidor proporcionó un certificado válido si requerido, además de revisar que los parámetros de *servidor_hola* son aceptados. Si todo es satisfactorio, el cliente envía uno o más mensajes de vuelta al servidor.
Si el servidor ha solicitado un certificado, el cliente empieza esta fase enviando un mensaje certificado. Si ningún certificado esta disponible, el cliente envía una alerta de *no_certificado* de inmediato.
El mensaje cliente_intercambio_llave debe ser enviado en esta fase, su contenido depende del tipo de intercambio de llave y puede ser mediante los cuatro métodos siguientes:

- RSA. El cliente genera una llave pre-maestra secreta de 48 bytes y la encripta con la llave pública del certificado del servidor o la llave temporal RSA del mensaje sevidor_intercambio_llave. Es usada para calcular el secreto maestro.
- Diffie-Hellman fijo o anónimo. Los parámetros públicos Diffie-Helmman del cliente son enviados.
- Diffie-Helman fijo. Los parámetros públicos Diffie-Hellman del cliente son enviados en un mensaje certificado, entonces el contenido del mensaje es nulo.
- Fortezza. Los parámetros e Fortezza son enviados.

Finalmente, en esta fase el cliente puede enviar un mensaje verificación_certificado para proporcionar la verificación explícita del certificado de cliente. Este mensaje es enviado seguido de cualquier certificado de cliente que tiene las capacidades de la firma. Este mensaje firma un código de dispersión basado en los mensajes anteriores y definido como de la siguiente manera:

CertificadoVerifiación.firma.md5_hash
MD5(secreto_maestro || bloque_2 || MD5 (negociación_mensaje || secreto_maestro || bloque_1));
Certificado.firma.sha_hash
SHA(secreto_maestro || bloque_2 || MD5 (negociación_mensaje || secreto_maestro || bloque_1));

Donde bloque_1 y bloque_2 son valores definidos en el MAC, el *mensaje_negociación* se refiere a todos los mensajes del Protocolo de Negociación enviados o recibidos empezando en el *cliente_hola* pero no incluye este mensaje y el *secreto_maestro*. Si la llave privada del usuario es DSS, entonces es usado para encriptar el compendio SHA-1. Si la llave privada del usuario es RSA, es usado para encriptar la concatenación de los compendios MD5 y SHA-1. En este caso, el propósito es verificar la llave privada del propio cliente para el certificado cliente. En el caso de que alguien esté empleando mal el certificado del cliente, no podrá enviar este mensaje.

Fase 4. Terminar

Esta fase completa el establecimiento de la conexión segura. El cliente envía un mensaje de *especificación_cambio_cifrado* y copia la especificación del algoritmo de cifrado pendiente en la actual. Especificación de cifrador. Este mensaje no es considerado en el Protocolo de Negociación pero es enviado usando el Protocolo de Especificación de Cambio de Cifrador. El cliente envía inmediatamente un mensaje de fin bajo el nuevo algoritmo, llaves y secretos. El mensaje final verifica que la llave de intercambio y el proceso de autentificación sean exitosas. El contenido del mensaje final es la concatenación de dos valores de compendio:

MD5(secreto_maestro || bloque_2 || MD5 (mensaje_negociación || Emisor || secreto_maestro || bloque_1));
SHA(secreto_maestro || bloque_2 || MD5 (mensaje_negociación || Emisor || secreto_maestro || bloque_1));

donde el *Emisor* es el código que identifica que el emisor es el cliente y los *mensaje_negociación* son todos los datos de mensaje de negociación pero que no incluyen el mensaje. En respuesta a estos mensajes, el servidor envía su propio mensaje *especificación_cambio_cifrado*, transfiere el pendiente a la actual especificación de cifrado y posteriormente envía su mensaje final. En este punto la negociación está terminada, entonces el cliente y el servidor pueden empezar a intercambiar datos de la capa de aplicación.

Creación del secreto maestro

El secreto maestro compartido es un valor de tiempo de 48 bytes (384 bits) generado en la sesión para ofrecer seguridad en el intercambio de llave. La creación es en dos fases. Primero un *pre_secreto_maestro* es intercambiado. Segundo, el *secreto_maestro* es calculado por los dos participantes. Para *pre_secreto_maestro*, existen dos posibilidades:

- RSA. Un *pre_secreto_maestro* de 48 bytes es generado por el cliente, encriptado con la llave pública RSA del servidor, y enviado a el servidor. El servidor desencripta el texto cifrado usando su llave privada para cubrir el pre_secreto_maestro.
- Diffie-Hellman. Tanto el cliente, como el servidor generan una llave pública Diffie-Hellman. Después, éstas son intercambiadas y cada lado ejecuta el cálculo Diffie-Hellman para crear el pre_maestro_secreto.

Ambos lados ahora calculan el secreto_maestro como sigue:

```
Secreto_maestro= MD5(pre_secreto_maestro || SHA('A'   || pre_secreto_maestro ||
                     ClienteHola.aleatorio || ServidorHola.aleatorio )) ||
                 MD5(pre_secreto_maestro  || SHA('BB'  || pre_secreto_maestro ||
                     ClienteHola.aleatorio || ServidorHola.aleatorio )) ||
                 MD5(pre_secreto_maestro  || SHA('CCC'|| pre_secreto_maestro ||
                     ClienteHola.aleatorio || ServidorHola.aleatorio ))
```

donde ClienteHola.aleatorio y ServidorHola.aleatorio son los actuales valores intercambiados al inicio de los mensajes hola.

4.11. Sistema de autentificación Kerberos.

El sistema de autentificación y de distribución de llaves Kerberos fue desarrollado por el Instituto Tecnológico de Massachusetts (MIT); se llamó así por un perro de varias cabezas de la mitología griega que solía cuidar la entrada al averno. El objetivo principal de Kerberos fue extender la noción de autorización, autentificación, y contabilidad de entorno computacional y de red del MIT.

Kerberos se basa en el modelo cliente-servidor. El objetivo de Kerberos es permitir que un cliente mediante el nombre de un usuario particular pruebe su identidad a un servicio, o al correspondiente servidor de aplicaciones, sin

necesidad de enviar datos por la red que podrían facilitar el que un atacante suplantase posteriormente al usuario. Para conseguir este objetivo, el modelo de Kerberos requiere la existencia de una tercera entidad de confianza que actúe como centro de distribución de llaves (KDC, Key Distribution Center) en el reino de Kerberos. El KDC está constituido de dos elementos:

- Un servidor de autentificación (AS, Authentication Server).
- Un conjunto de servidores de emisión de billetes (TGS, Ticket Granting Servers).

El KDC mantiene una base de datos con una entrada por cada usuario registrado en el reino. Para permitir que la seguridad de los datos sea la principal consideración cuando existen compromisos operacionales en la gestión de Kerberos, la información que Kerberos almacena es la mínima necesaria para realizar y gestionar las tareas de autentificación.

Kerberos cifra todas las llaves maestras de los usuarios con una llave maestra del KDC. Este cifrado permite que el administrador del sistema elimine copias de la base de datos del KDC desde el servidor maestro y envíe copias a los servidores esclavos sin requerir longitudes extraordinarias para proteger la privacidad de las copias. Los servidores esclavos son necesarios en reinos grandes para proporcionar un servicio de seguridad con una disponibilidad adecuada. Lo importante es que Kerberos no almacena la llave maestra del KDC en la misma base de datos, sino que gestiona la llave de forma separada.

Kerberos proporciona varias herramientas para la gestión de la base de datos del KDC. En reinos pequeños, se puede utilizar un conjunto de herramientas manuales desde la consola del administrador del sistema. Para reinos más grandes, existe un servicio de automatizado de gestión del sistema (SMS, System Management Service). En cada caso, las funciones de gestión de Kerberos se realizan de forma remota por la red, y Kerberos autentifica las conexiones a la base de datos del KDC. Las actualizaciones se realizan mediante un protocolo que se ejecuta entre un cliente autentificado en una estación de trabajo y la base de datos del KDC (Figura 4.31). Si la base de datos del KDC no está accesible, las actualizaciones se inhabilitan de forma temporal.

Kerberos utiliza billetes para distribuir las llaves de sesión. De forma general, un billete es un registro de datos que se puede utilizar para la autentificación. En Kerberos, un billete consiste en un certificado emitido por el KDC de Kerberos y cifrado con la llave maestra del servidor. Entre otra información, el billete contiene:

- La llave de sesión que se utilizará para la autentificación entre el cliente y el servidor.
- El nombre del principal al que se ha enviado la llave de sesión.
- Un tiempo de expiración tras el que se considera que la llave de sesión ya no es válida.

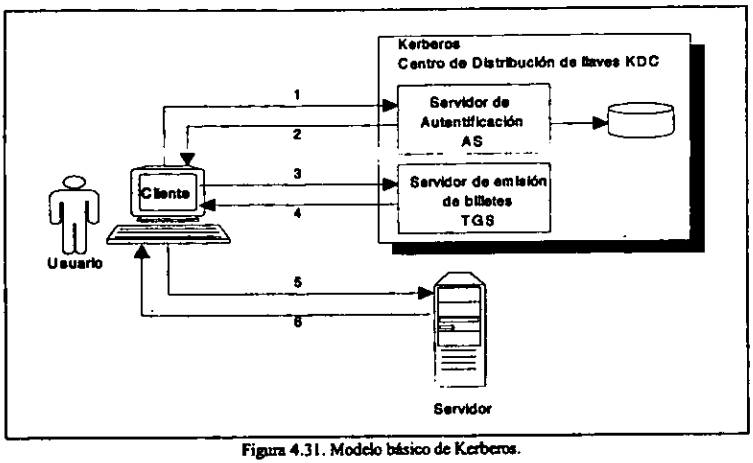


Figura 4.31. Modelo básico de Kerberos.

El billete no se envía directamente al servidor, sino que se envía al cliente, el cual, a su vez, lo reenvía al servidor como parte del intercambio para la autentificación. Un billete de Kerberos siempre está cifrado con la llave del servidor, conocida sólo por el AS y el servidor en cuestión. Debido a este cifrado, el cliente no puede modificar el billete sin ser detectado.

4.11.1. Kerberos V4.

La versión V4 de Kerberos es un protocolo que se caracteriza por el uso de marcas temporales; servicio de emisión de billetes para dar soporte a las subsiguientes operaciones de autentificación sin necesidad de hacer que el usuario introduzca constantemente su contraseña; además, ofrece una solución para la autentificación entre reinos. El procedimiento de este protocolo se observa en la figura 4.31 y se describe en la siguiente tabla.

1:	C	→ AS	: U, TGS, T, L
2:	AS	→ C	: $\{K, N, T_{c,tgs}\}K_U$
3:	C	→ TGS	: S, N, $T_{c,tgs}$, $A_{c,tgs}$
4:	TGS	→ C	: $\{T_{c,s}, K'\}K$
5:	C	→ S	: $T_{c,s}$, $A_{c,s}$
6:	S	→ C	: $\{T+1\}K'$

Tabla 4.22. Seis pasos del protocolo Kerberos V4.

Los seis pasos en el que consiste el protocolo de Kerberos se pueden agrupar en tres intercambios:

- El intercambio con AS, entre el cliente y el AS (pasos 1 y 2).
- El intercambio con TGS entre el cliente y el TGS (pasos 3 y 4).
- El intercambio con AP entre el cliente y el servidor de la aplicación (pasos 5 y 6).

El intercambio con AS

Cuando un usuario va a una estación de trabajo e intenta entrar, debe proporcionar el nombre de usuario *U*. Posteriormente, el cliente *C* utiliza un servicio de nombres para obtener una lista de *TGS* disponibles en ese momento, y selecciona el más cercano en términos de topología de red. *C* envía entonces un mensaje KRB_AS_REQ (solicitud de servidor de autentificación de Kerberos) al *AS* (paso 1). El mensaje contiene el identificador de usuario *U*, el identificador del *TGS* seleccionado, una marca temporal *T* y el tiempo de vida deseado para el TGT (Ticket Granting Ticket, billete de concesión de billete), *L*.

Después de recibir el mensaje KRB_AS_REQ, el AS busca y extrae la llave secreta asociada con *U* y el *TGS*. El *AS* selecciona entonces de forma aleatoria una nueva llave de sesión *K*, y devuelve un mensaje KRB_AS_REP (respuesta del servidor de autentificación de Kerberos) a *C* en el paso (2). El mensaje contiene *K*, *N*, y $T_{c,tgs}$ = { U, C, TGS, T, L, K}K_{tgs}. Este mensaje está cifrado con la llave del usuario K_U.

La *N* corresponde a un valor temporal que sirve para que *C* corrobore que el mensaje se trata de una réplica del AS a la solicitud actual. El término $T_{c,tgs}$ = {U, C, TGS, T, L, K}K_{tgs} se utiliza para indicar un TGT emitido por el AS que es utilizado por el cliente C para autentificarse ante el TGS y solicitar el correspondiente billete.

Al recibir *C* el mensaje KRB_AS_REP, éste pide al usuario *U* que introduzca su contraseña, si *U* introduce correctamente su contraseña, *C* puede utilizar una función de hash unidireccional *h* para calcular la llave secreta K_U = $h(pwd_U)$. Con esta llave, C puede descifrar {K, N, $T_{c,tgs}$ }K_U y extraer en consecuencia K, N, y $T_{c,tgs}$. Si C tiene éxito poseerá un TGT que podrá utilizar para solicitar billetes al TGS.

En un TGT, el campo de tiempo de vida se utiliza como si fuera un tiempo de expiración de la contraseña. Al limitar el tiempo de vida de un TGT, se limita la cantidad de daño que se puede hacer en el caso de que un TGT se vea comprometido. En Kerberos, generalmente no existe la posibilidad de revocar un TGT una vez que ha sido emitido. Por tanto, al limitar el tiempo de vida del TGT, implícitamente se establece un instante a partir del cual el TGT se considera obsoleto.

El intercambio con TGS

El formato de los mensajes que se intercambian entre el cliente *C* y el *TGS* es prácticamente idéntico al del intercambio con AS. La principal diferencia es que el cifrado y descifrado del intercambio con *TGS* no se realiza con la llave de usuario K_U, sino bajo la llave de sesión *K* que *C* comparte con el *TGS*.

Antes de iniciar el intercambio con *TGS*, *C* debe determinar en qué reino está registrado el servidor al que le va a solicitar el billete. Si *C* no posee un *TGT* para ese reino, deberá obtener uno. En primer lugar, se intenta solicitar al

servidor local de Kerberos un *TGT* para el reino de destino (utilizando el mensaje KRB_TGS_REQ). El servidor de Kerberos puede entonces devolver el *TGT* para el reino deseado, en cuyo caso *C* puede proceder. Puede suceder también que Kerberos devuelva un *TGT* para un reino que está más cercano al deseado. En este caso, este paso se deberá repetir utilizando el servidor de Kerberos del reino especificado en el TGT. Si no se devuelve ningún billete, el intento deberá repetirse utilizando un servidor de Kerberos del reino de jerarquía superior. Esta solicitud necesita, a su vez, un TGT para el reino de jerarquía superior, que se debe obtener, a su vez, de forma recursiva. Una vez que el cliente obtiene el TGT del reino apropiado, determina qué servidor de Kerberos se ocupa de ese reino y contacta con él. La lista puede obtenerse mediante un archivo de configuración o mediante el correspondiente servicio de red.
Una vez que C posee el TGT, envía al *TGS* un mensaje KRB_TGS_REQ (solicitud de servidor de emisión de billetes de Kerberos) en el paso (3). El mensaje contiene *S*, *N*, $T_{c,tgs}$ y un autentificador $A_{c,tgs}$ = {C, T}K, donde, *S* es el nombre del servidor solicitado y *T* es una marca temporal. El autentificador es un registro de datos con información que ha sido cifrada con la llave de sesión *K* conocida sólo por el cliente y el servidor requerido. El propósito del autentificador es demostrar que *C* posee la llave secreta y evitar que el $T_{c,tgs}$ pueda ser replicado, mediante la marca temporal.
La marca temporal representa el número de segundos desde las 00:00:00 GMT (hora de Greenwich) del 1 de enero de 1970, y está codificada con cuatro bytes. Análogamente, el tiempo de vida está especificado en unidades de cinco minutos y codificado con un byte. Por lo tanto, el tiempo de vida máximo que se puede expresar en la versión 4 de Kerberos es 2^8*5= 1280 minutos, ligeramente superior a 21 horas.
Si el TGS considera que el billete y el autentificador son válidos, en el paso (4) devuelve un mensaje KRB_TGS_REP (respuesta del servidor de emisión de billetes de Kerberos), el cual incluye un billete $T_{c,s}$ y una nueva llave de sesión K', ambos cifrados con K El término $T_{c,s}$ = {U, C, S, T', L', K'}K_S es utilizado por C para autentificarse ante el servidor S, donde, T' indica una nueva marca temporal y L' indican un nuevo tiempo de vida del billete. Cuando C recibe el mensaje KRB_TGS_REP, lo descifra con la llave de sesión compartida con el TGS.

El intercambio con AP

El intercambio con AP es utilizado por las aplicaciones de red para autentificar un cliente ante un servidor, o para la autentificación mutua entre un cliente y un servidor. El cliente debe haber adquirido credenciales para el servidor utilizando los intercambios con AS o con TGS.
En el paso (5), C envía al servidor S un mensaje KRB_AP_REQ (solicitud de aplicación de Kerberos). El mensaje contiene $T_{c,s}$, $A_{c,s}$ = {C, T'}K' e información adicional de contabilidad. La autentificación se basa en la cuenta de tiempo del servidor, el billete $T_{c,s}$ y el autentificador $A_{c,s}$.
Para asegurarse de que el mensaje KRB_AP_REQ no es una réplica de una solicitud lo bastante cercana en tiempo como para entrar dentro del margen temporal, S debe mantener todas las marcas temporales recibidas dentro del máximo margen temporal permitido, y comprobar que cada marca temporal recibida es diferente de cualquiera de los valores almacenados. Si el autentificador es más antiguo que el máximo margen temporal permitido es rechazado. Si no se producen errores, y si se requiere autentificación mutua, en el paso (6) S debe devolver a C un

mensaje KRB_AP_REP (respuesta de aplicación de Kerberos). Este mensaje se cifra de nuevo con la llave de sesión K', compartida por C y S. Como esta llave estaba en el billete cifrado con la llave secreta del servidor, la posesión de dicha llave prueba que S es el principal deseado. Más concretamente, S debe incrementar la marca temporal incluida en el autentificador del mensaje KRB_AP_REQ y volver a cifrarla con K'.

Autentificación entre reinos

Un entorno de red puede formarse a partir del agrupamiento de varias organizaciones, como empresas en competencia mutua, agencias gubernamentales, instituciones financieras, o universidades. En un entorno como ese, sería difícil encontrar una entidad de alta fiabilidad y confianza, ya que el problema es, quien ejecuta y gestiona el KDC puede acceder a cualquier llave maestra de usuario, y por lo tanto puede acceder a todo lo que el usuario pueda acceder. Incluso si hubiera una entidad en la que todo el mundo confiara, no es suficiente ya que estaría muy ocupada, tendría que procesar todas las realizaciones de los usuarios y de los servicios que entran y salen del entorno de red.

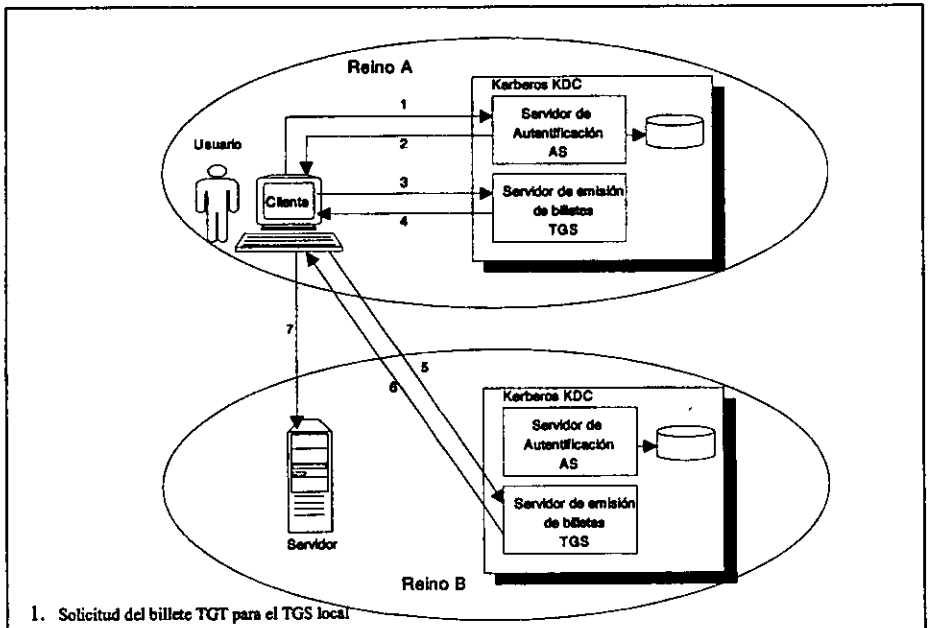


Figura 4.32. Solicitud de servicios en otro reino de Kerberos

Por esta razón, los entornos de red se dividen en reinos. Cada reino posee y ejecuta su propio KDC. En un reino puede haber varios KDC, pero serían equivalentes, poseerían la misma llave maestra y tendrían las mismas entradas en la base de datos para los usuarios. Sin embargo, dos KDC de diferentes reinos tendrían llaves maestras distintas, y bases de datos de llaves maestras de usuarios completamente diferente.

En Kerberos, la autentificación entre reinos consiste en autentificar un cliente de un reino ante un servidor de otro reino. Una llave entre reinos es una llave secreta compartida entre los KDC de dos reinos de Kerberos. Al establecer las llaves entre reinos, los administradores de los reinos de Kerberos permiten que un cliente autentificado en un reino pueda utilizar remotamente su autentificación. Como se observa la figura 4.32, el intercambio de llaves entre reinos registra al TGS de cada reino como un principal del otro reino. Los clientes pueden entonces obtener TGT para el reino remoto desde el TGS del reino local. Cuando se utilizan esos TGT, el TGS remoto utiliza la llave entre reinos para descifrar el TGT, y certificar así que ha sido enviado por el TGS del cliente. Los billetes enviados por el TGS remoto indicarán a un servicio que el cliente ha sido autentificado en otro reino.

Se dice que un reino se comunica con otro si dos comparten una llave entre reinos, o también si el reino local comparte una llave entre reinos con un reino intermedio que se comunica con el reino remoto. En terminología de Kerberos, se define vía de autentificación como la secuencia de reinos intermedios que se cruzan en el proceso de autentificación cuando un reino se comunica con otro.

4.11.2. Kerberos V5.

La versión V4 de Kerberos se desarrolló inicialmente para el entorno de computación y de red del proyecto Athena, y las soluciones propuestas están pensadas para ese entorno y no siempre se pueden aplicar con éxito en otro ambiente. La versión 5 de Kerberos esta especificada en el RFC 1510 y proporciona un número de mejoras sobre la versión 4. A continuación se mencionan las deficiencias de entorno y las deficiencias técnicas de la versión 4, que reemplazaron para dar origen a la versión 5 de Kerberos

Deficiencias de entorno

- En la versión 5, el cifrado es marcado con un identificador que indica el tipo de cifrado, por lo que, cualquier técnica de cifrado puede ser usada. Con el identificador, el receptor se entera que algoritmo de cifrado debe utilizar para descifrar el mensaje. Las llaves de cifrado son etiquetadas con el tipo de cifrado y la longitud, permitiendo que la misma llave sea usada en diferentes algoritmos.
- La versión V4 de Kerberos requiere el uso de direcciones de protocolo Internet (IP), lo que lo hace inadecuado en algunos entornos. Por otra parte, las direcciones de red en la versión 5 son identificadas con un tipo y una longitud de campo, de forma que el receptor pueda interpretarlos adecuadamente. Si un sistema admite múltiples protocolos de red o tiene múltiples direcciones de un solo tipo, en un billete se pueden proporcionar todos los tipos y direcciones.

- En la versión V4 de Kerberos, el sistema emisor codifica los mensajes de red que ocupan varios bytes utilizando su propia ordenación de bytes. Por otra parte, el receptor realiza la conversión de esta ordenación en su propia ordenación nativa. Aunque esto simplifica la comunicación entre dos sistemas con la misma ordenación de bytes, no establece convenciones fijas y hace imposible la operación de máquinas con ordenaciones de bytes no usuales, que no son entendidas por el receptor.

  En la versión 5, todos los mensajes estructurados son definidos usando ANSI.1(Abstract Syntax Notation One) y BER (Basic Encoding Rules).
- Como anteriormente se mencionó, el máximo tiempo de vida que se puede expresar en los billetes de la versión 4 de Kerberos es ligeramente superior a 21 horas. Este límite es una de las principales deficiencias de la versión V4 de Kerberos, ya que imposibilita la asignación de billetes a tareas de larga duración. En la versión 5 de Kerberos, los billetes se pueden emitir con la característica de renovables, asociándoles dos tiempos de caducidad: uno de ellos en un futuro cercano, y el otro posterior. El billete expira en el primero de esos tiempos en la forma habitual, pero si antes de la fecha de caducidad se le presenta al TGS una solicitud de renovación, se puede devolver un billete de reemplazo que es válido durante un intervalo de tiempo adicional. Pero el TGS no renovará el billete más allá del segundo instante de expiración indicado en el mismo. Este mecanismo tiene la ventaja de que aunque las credenciales se pueden utilizar durante periodos prolongados de tiempo, el TGS puede rechazar la renovación de billetes marcados como robados, evitándo así su utilización continuada.
- En la versión 4 de Kerberos, los usuarios se nombran utilizando tres componentes: nombre, realización y reino. Cada uno de ellos puede tener una longitud de hasta 40 caracteres. Sin embargo, estos tamaños han resultado ser cortos en algunas aplicaciones y entornos de instalación. En Kerberos V5, un identificador de usuario consta de dos partes: el nombre de reino y el resto. El nombre de reino está separado para facilitar la realización de rutinas que involucren varios reinos y comprobaciones de acceso dependientes del reino. El resto es una secuencia de tantos componentes como sean necesarios para nombrar al usuario.
- La versión 4 de Kerberos permite que los reinos cooperen mediante la compartición de una llave secreta. Un determinado cliente puede obtener billetes para servicios en otros reinos, consiguiendo primero un TGT para el reino externo a través de su KDC local, y utilizando después el TGT para obtener el billete en el servidor de aplicaciones del reino externo. Este intercambio de billetes por parejas hace que las solicitudes de billetes entre reinos sean fáciles de realizar, pero la interconexión de n reinos requiere n^2 llaves. Incluso cuando sólo hay unos pocos reinos interconectados, la asignación y gestión de este número de llaves entre reinos es una tarea costosa. Por el contrario, la versión 5 admite la autentificación multipaso entre reinos, de forma que las llaves se puedan compartir jerárquicamente. Con la versión 5 de Kerberos, cada reino comparte llaves con su reino padre y sus reinos hijos. Si dos reinos no comparten directamente una llave entre reinos, la organización jerárquica permite que se establezca una vía de autentificación. Si no se utiliza una organización jerárquica, puede que sea necesario consultar alguna base de datos para construir una vía de autentificación entre reinos. Aunque los reino se organizan típicamente de forma jerárquica, los reinos intermedios se puedan saltar para realizar la autentificación entre reinos mediante vías alternativas de autentificación. Esto se podría hacer para conseguir una comunicación

más eficiente entre dos reinos. Para facilitar esta comunicación, existe un campo en cada billete que contiene la secuencia de los reinos atravesados, excluyendo los reinos fuente y destino.

Deficiencias técnicas

- En la versión 4 de Kerberos, el TGT emitido por el AS está cifrado dos veces cuando vuelve al cliente, y sólo una vez cuando se envía al TGS. En general, no hay ninguna necesidad de cifrar dos veces un billete. Hacerlo así puede representar un gasto inútil de potencia de cálculo si la operación de cifrado es costosa.
- La versión 4 de Kerberos utiliza una versión no estándar de DES, conocida encadenamiento de bloque cifrado y plano (PCBC, plain and cipher block chaining). En el modo PCBC, un bloque de texto sin cifrar P_{i+1}, antes de ser cifrado es combinado mediante la operación OR exclusivo con el texto sin cifrar P_i y con el texto cifrado C_i. Este modo posee la propiedad de que al ser modificado un bloque de texto cifrado C_i se modifican todos los bloques de texto descifrado desde el P_i en adelante. La versión 4 de Kerberos coloca unos datos reconocibles al final del mensaje que va a cifrar, de forma que se pueda reconocer si el bloque final se ha descifrado adecuadamente. Se supone que si el bloque final se descifra correctamente, entonces los datos no han sido alterados entre su transmisión por la fuente y su recepción en el destino. Sin embargo, el uso de DES en modo PCBC y la comprobación de los contenidos sólo en el bloque final descifrado no garantiza que Kerberos sea capaz de detectar todas las corrupciones de mensajes. Por ejemplo, si un intruso puede intercambiar dos bloques adyacentes, PCBC se resincroniza tras esos dos bloques, y el bloque final será descifrado adecuadamente. La versión 5 utiliza el modo CBC estándar para el cifrado, añadiendo al texto sin cifrar una suma de comprobación para verificar que el texto cifrado no ha sido alterado durante su recorrido.
- Llaves de sesión: Cada billete incluye una llave de sesión que es usada por el cliente para cifrar el autentificador con el billete. Pero la llave de sesión puede ser usada subsecuentemente por el cliente y el servidor para proteger los mensajes enviados durante la sesión. Sin embargo, el mismo billete puede ser usado repetidamente para obtener un servicio de un servidor particular, por lo que, existe el riesgo de que un intruso repitiera los mensajes desde una sesión antigua al cliente o al servidor. En la versión 5, es posible para el cliente y el servidor negociar otra llave de sesión, la cual es usada únicamente para una conexión. Una nueva solicitud de acceso del cliente resultaría el uso de una nueva llave de sesión.
- Ataque a contraseñas: Ambas versiones son vulnerables a los ataques a contraseñas. El mensaje del AS a los clientes incluye material cifrado con una llave basada en la contraseña del cliente. Si un intruso captura este mensaje lo descifra adecuadamente, descubrirá la contraseña y posiblemente obtenga credenciales de autentificación de Kerberos. La versión 5 proporciona un mecanismo conocido como preautentificación que hace los ataques de contraseña más difíciles pero no los previene totalmente.

Funcionamiento de Kerberos V5

Kerberos se ocupa fundamentalmente de la autentificación. No se ocupa directamente de los servicios relacionados con la seguridad, como las autorizaciones o la contabilidad. Para admitir las implantaciones de estos servicios de seguridad, la versión 5 de Kerberos proporciona un mecanismo para la transmisión de información de autorización y de contabilidad a prueba de alteraciones como parte de un billete. Esta información puede restringir el uso del billete. La codificación de la correspondiente restricción no es asunto del protocolo de Kerberos, y se encontrará definida por los mecanismos de autorización y contabilidad que se utilicen. Las restricciones se codifican en el campo de datos de autorización del billete.

Cuando se solicita un billete, las restricciones se especifican y se envían al TGS, donde se insertan en el billete, se cifran y se protegen contra alteraciones. En la forma más general del protocolo, un cliente solicita que el KDC incluya o añada esos datos en un nuevo billete. El TGS no elimina del TGT ningún dato sobre autorización, sino que los copia en cada nuevo billete, y añade también los datos adicionales sobre autorización que se requieran. Tras el descifrado del billete, los datos de autorización estarán también disponibles para el servidor de aplicaciones. Aunque Kerberos no realiza ninguna interpretación de los datos, espera que el servidor de aplicaciones utilice los datos de autorización para restringir el acceso del cliente al recurso especificado en el billete.

Entre otros usos, el campo de datos de autorización puede ser también utilizado por un billete de sustitución para crear una capacidad. El cliente que solicita el billete de sustitución a un TGS especifica las restricciones en el campo de datos de autorización y seguidamente transmite de forma segura la llave de sustitución y de sesión a la otra parte, que utiliza el billete para obtener el servicio limitado del servidor de aplicaciones.

En la versión 5, las aplicaciones pueden escoger entre utilizar la marca temporal como antes, o utilizar un número de secuencia. Si se utiliza una marca temporal, el receptor debe almacenar las marcas temporales conocidas para evitar los ataques de réplica. Si se utiliza alternativamente un número de secuencia, entonces el receptor debe verificar que los mensajes llegan en el orden apropiado sin saltos.

Los pasos del protocolo de la versión 5 de Kerberos se pueden formalizar como sigue:

1:	C	→ AS	: U, TGS, L_1, N_1
2:	AS	→ C	: $U, T_{c,tgs}, \{TGS, K, T_{inicio}, T_{fin}, N_1\}K_U$
3:	C	→ TGS	: $S, L_2, N_2\ T_{c,tgs}, A_{c,tgs}$
4:	TGS	→ C	: $U, T_{c,s}, \{S, K', T'_{inicio}, T'_{fin}, N_2\}K$
5:	C	→ S	: $T_{c,s}, A_{c,s}$
6:	S	→ C	: $\{T'\}K'$

Tabla 4.23. Seis pasos del protocolo Kerberos V4.

A igual que en la versión 4 de Kerberos, $T_{c,tgs}$ y $T_{c,s}$ se utilizan para indicar un TGS y un billete respectivamente. $A_{c,tgs}$ y $A_{c,s}$ se utilizan para indicar los autentificadores correspondientes.

En el paso (1), C selecciona un TGS, y envía el mensaje KRB_AS_REQ al AS de Kerberos. El mensaje contiene el identificador de usuario U, el identificador del TGS seleccionado, un tiempo de vida solicitado para el TGT, L_t y un valor efímero N_t. Además, el mensaje puede especificar opciones como:

- Si se va a realizar preautentificación.
- Si el billete seleccionado va a ser renovable, sustituible o reenviable.
- Si la validez está retrasada o se va a permitir la validez retrasada de los billetes derivados.
- Si se aceptará un billete renovable en lugar de un billete no renovable en el caso de que la fecha de caducidad del billete no se pueda satisfacer utilizando un billete no renovable.

Después de que el AS recibe el mensaje KRB_AS_REQ, busca y extrae las llaves secretas de U y del TGS. Si se requiere, el AS preautentifica la solicitud, y si falla la preautentificación, le devuelve a C un mensaje de error. De lo contrario, el AS selecciona aleatoriamente una nueva llave de sesión K, y le envia a C un mensaje KRB_AS_REP en el paso (2). El mensaje contiene U, un TGT $T_{c,tgs} = \{U, C, TGS, K, T_{inicio}, T_{fin}\}K_{tgs}$ y $\{TGS, K, T_{inicio}, T_{fin}, N_t\}K_U$. Los tiempos de inicio y de expiración del TGT, T_{inicio} y T_{fin}, se establecen de acuerdo con la política de seguridad del reino, de forma que posiblemente se ajuste al tiempo de vida L_t especificado en el mensaje KRB_AS_REQ.
Tras recibir el mensaje KRB_AS_REP, C aplica la función de hash unidireccional h a la contraseña suministrada por el usuario, pwd_U para calcular la llave maestra $Ku = h(pwd_U)$. Obtenida esta llave, C puede descifrar $\{TGS, K, T_{inicio}, T_{fin}\}K_U$ y extraer consecuentemente TGS, K, T_{inicio}, T_{fin}. Ahora, C posee un TGT que es válido desde T_{inicio} hasta T_{fin}, y puede utilizar el TGT para solicitar billetes al TGS.
El intercambio con TGS se inicia siempre que C desea obtener un billete de un servidor, renovar o validar un billete existente o incluso obtener un billete de intercambio. Al menos en el primer caso, el cliente debe haber obtenido ya un TGT mediante un intercambio con AS. En el paso (3), C envía al TGS un mensaje KRB_TGS_REQ que contiene información con la autentificación del cliente y una solicitud de credenciales. La información de autentificación consiste de el $T_{c,tgs}$ del TGT y el correspondiente autentificador $A_{c,tgs}$ $\{C, T\}K$ que C genera con una marca temporal T, utilizando la llave de sesión K En el paso (4), el TGS devuelve un mensaje KRB_TGS_REP que contiene un billete $T_{c,s} = \{U, C, S, K', T'_{inicio}, T'_{fin}\}K_S$ para el servidor solicitado, así como $\{S, K', T'_{inicio}, T'_{fin}\}K$.
En el intercambio con AP existen dos formas de autentificación, una es autentificar un cliente ante un servidor o la autentificación mutua entre cliente y servidor. En el paso (5), C envía al servidor S el mensaje KRB_AP_REQ, que contiene el billete $T_{c,s}$ y el correspondiente autentificador $A_{c,s}$ $\{C, T'\}K'$, junto con alguna información de contabilidad. La autentificación se basa en la hora del día actual que tienen el servidor, el autentificador y el billete. Si no hay errores y se requiere autentificación mutua, S devuelve un mensaje KRB_AP_REP en el paso (6). Este mensaje contiene la marca temporal T', cifrada con la llave de sesión K' que C y S comparten en ese momento.

4.12. Firewalls.

Los firewalls, también conocidos como muros de seguridad, son otra alternativa para proteger el tránsito de información. Su principal tarea es la de examinar todo el tráfico de entrada y salida, permitiendo el paso solamente al tráfico autorizado. Originalmente, los firewalls se enfocan principalmente sobre control de servicios, pero ellos se han involucrado para proporcionar cuatro:

- Control de servicio. Determina los tipos de servicios de Internet que pueden ser accesados, verificando las direcciones IP o los números de puerto TCP.
- Control de dirección. Mediante la dirección de las peticiones de servicios particulares, determina si éstas pueden ser iniciadas y transmitidas a través del firewall.
- Control de usuario. Controla el acceso a un servicio de acuerdo al usuario que intente accesarlo.
- Control de comportamiento. Controla como son usados los servicios particulares. Por ejemplo, el firewall puede filtrar el correo electrónico para eliminar spams.

Los firewalls usan tres tecnologías diferentes: filtro de paquetes, gateways a nivel de circuitos y gateways a nivel aplicación.

4.12.1. Filtro de Paquetes.

El filtro de paquetes es transparente al usuario y puede ser instalado como parte de ruteador, de tal forma que, los paquetes que satisfacen algún criterio se reenvían y los que fallan se descartan. Este esquema se observa en la figura 4.33. El ruteador trabaja en las capas de red (protocolo IP) y de transporte(protocolo TCP ó UDP) del modelo OSI. Casi todos los dispositivos de filtro de paquetes operan de la siguiente forma. Las reglas de filtrado de paquetes debe almacenarse en los puertos del dispositivo. Cuando el paquete llega al puerto, entonces se analizan los encabezados de los paquetes IP y TCP o UDP, incluyendo los campos de la dirección IP fuente y destino, el campo de protocolo de IP y el número de puerto TCP ó UDP.

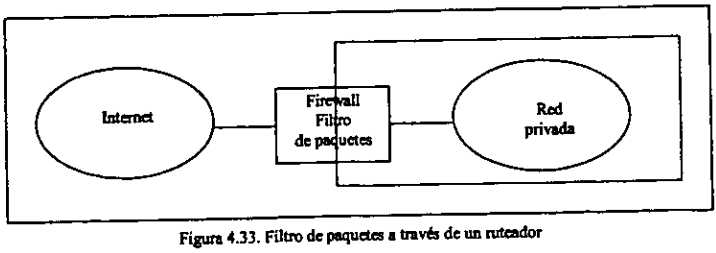


Figura 4.33. Filtro de paquetes a través de un ruteador

Las reglas de filtrado de paquetes se almacenan en un orden específico. Cada regla se aplica al paquete en el mismo orden en el que se almacenó. Si una regla permite la transmisión o recepción de un paquete, entonces se le permite proceder. Pero, si una regla bloquea la transmisión o recepción de un paquete, el paquete no se acepta. Por otra parte, si un paquete no satisface ninguna regla, es bloqueado.

La lista de filtros se aplican secuencialmente, de forma que la primera regla que el paquete cumpla marcará la acción a realizar (descartarlo o dejarlo pasar). La aplicación de las listas de filtros se puede hacer en el momento de entrada del paquete o bien en el de salida o en ambos.

Uno de los ataques utilizados con más frecuencia para saltarse la protección establecida por un firewall es el 'address-spoofing', que consiste en generar paquetes IP con direcciones de origen falsas.

Los filtradores de paquetes son una buena solución, pero tienen sus limitaciones a la hora de tratar los servicios como tales, pues para ellos cada paquete es independiente y no forma parte de ningún todo, por lo tanto, de ningún servicio. Además, existen servicios como DNS o FTP, que dificultan realizar una configuración segura de un filtrador de paquetes.

4.12.2. Gateways a nivel de aplicación.

La principal función de los gateways a nivel aplicación es examinar los contenidos de cada paquete. Un getway también es conocido con el nombre de pasarela.

Una pasarela a nivel aplicación, también llamado servidor proxy, actúa como un transmisor de tráfico a nivel aplicación (Figura 4.34). El usuario se comunica con la pasarela usando una aplicación TCP/IP, como el Telnet o FTP, y la pasarela pide al usuario su nombre e información de autentificación (Figura 4.34). La pasarela se comunica con la aplicación del anfitrión remoto y transmite segmentos TCP que contienen la aplicación entre los dos extremos. Si la pasarela no implementa el código proxy para la aplicaciones específicas, el servicio no es soportado y no puede ser remitida a través de la pasarela. La pasarela examina tanto los paquetes de salida como los de entrada. Esto elimina la posibilidad de que usuarios internos accedan a servicios no autorizados que puedan crear un agujero en la seguridad.

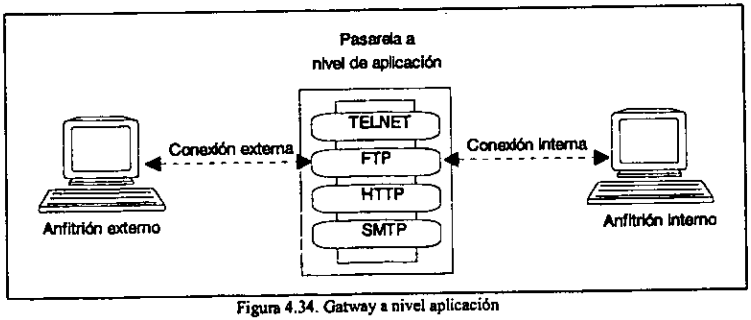


Figura 4.34. Gatway a nivel aplicación

Las pasarelas a nivel aplicación tienden a ser más seguras que los filtros de paquetes. Más que tratar de resolver las reglas de filtrado, la pasarela a nivel aplicación necesita únicamente examinar algunas aplicaciones permitidas. Una principal desventaja de este tipo de pasarela es la carga adicional de procesamiento en cada conexión.

4.12.3. Gatways a nivel de circuito.

Como se observa en la figura 4.35, todo el tráfico de entrada y salida está gobernado por un gateway o pasarela. La pasarela establece dos conexiones TCP, una entre ella misma y el anfitrión interno, y la otra entre ella misma y el anfitrión externo. Una vez que las dos conexiones son establecidas, la pasarela empieza a transmitir segmentos TCP desde una conexión a otra sin examinar el contenido. La función de seguridad consiste en determinar cuales conexiones serán permitidas. Una puerta de la pasarela actúa como un agente para sus usuarios. Este agente verifica las transmisiones y las acepta dependiendo del usuario. La pasarela entrega luego los datos a la dirección apropiada de su red interna. Además, la dirección del agente es la única información transmitida externamente en los paquetes de salida. También se tiene la ayuda de los proxies, los cuales, ayudan a limitar la cantidad de información de máquinas individuales que se enseña al mundo exterior.

14.13. Software de seguridad.

Además de los algoritmos, protocolos y sistemas vistos anteriormente, existen otras implementaciones comerciales (Com) y públicas (Fr), que ofrecen algunos mecanismos de seguridad sobre plataformas Windows (W) o Unix (U). A continuación se describen las características de estas herramientas que podemos encontrar en Internet.

Software	Objetivo	Descripción y características
CRACK (Fr-U)	Permite verificar el archivo de contraseñas de UNIX y encontrar palabras de paso triviales o poco seguras.	Comprueba a partir de reglas y de diccionarios, las palabras de paso que se encuentran en el archivo de contraseñas, creando un archivo con todos los usuarios y palabras descubiertas. Después realiza una serie de pasadas sobre el archivo de contraseñas aplicando la secuencia de reglas que se especifique. Estas reglas se encuentran en dos archivos (gecos.rules y icts.rules) y pueden ser modificadas. El algoritmo de cifrado que utiliza es Data Encryption Standard (DES).
PGP (Pretty Good Privacy) (Fr-U-W)	Utiliza criptografía de llave pública para proteger el correo electrónico y los archivos de datos.	Se basa en la criptografía de llave pública, donde es necesario la adquicisión de dos llaves complementarias llamadas: llave pública y llave privada. La versión 5.5 de PGP crea llaves usando encriptación Diffie-Hellman y el estándar de firmas digitales DSS. Esta versión, ya no genera llaves RSA para decifrar y verificar mensajes y archivos encriptados o firmados. El tamaño de la llave corresponde al número de bits usados para construir la llave digital. Existen dos opciones para seleccionar el tamaño de la llave, una de ellas especifica un rango de 768 a 3072 bits; mientras que la otra opción indica un rango de 512 a 4096 bits. También se puede seleccionar diferentes algoritmos de encriptación para la creación de llaves PGP, entre los cuales están: CAST, IDEA o Triple-DES. Estos tres algoritmos operan sobre bloques de texto no cifrado y cifrado de 64 bits. El tamaño de las llaves en IDEA y CAST es de 128 bits, mientras Triple-DES usa llaves de 168 bits.

Tabla 4.22. Software de seguridad.

Software	Objetivo	Descripción y características
GnuPG (Fr-U)	GnuPG está destinado a la plataforma UNIX, por lo que se hace imprescindible disponer de ese sistema operativo, cuya variante más popular para PC es Linux.	Es programa alternativo al PGP tradicional que permite cifrar archivos y mensajes de correo electrónico, de forma que sólo pueda acceder a ellos quien el usuario determine. También permite realizar un firmado digital de archivos y mensajes, lo que permite garantizar la integridad y la no-alteración de los mismos. A diferencia de PGP, GnuPG no utiliza algoritmos patentados, por lo que su uso es absolutamente libre y su licencia totalmente gratuita para cualquier uso (también comercial). Su código es abierto y puede ser libremente obtenido, examinado y compilado.
COPS. Computer Oracle and Password System (Com-U)	Es una de las herramientas de seguridad en Unix, y consiste de un conjunto de programas que revisan diferentes aspectos de la seguridad del sistema, reportando los posibles huecos de seguridad encontrados.	Existen dos versiones de este paquete: una versión escrita en "sh" y "C" y otra versión escrita en "perl", aunque su funcionalidad es similar. Entre las funcionalidades que tiene COPS destacan: ➢ Revisión de modos y permisos de los archivos, directorios y dispositivos. ➢ Palabras de paso pobres ➢ Verificación del contenido, formato y seguridad de los archivos de "password" y "group". ➢ Revisión de programas con root-SUID. ➢ Permisos de escritura sobre algunos archivos de usuario como ".profile" y ".cshrc" ➢ Configuración de ftp "anonymous".
Nessus (Fr-U)	Es un scanner de seguridad, encargado de realizar la auditoría a una determinada red y mostar si existen atacantes que desean irrumpir o hacer mal uso de ella.	Nessus mediante un planteamiento cliente-servidor, desarrolla una exploración exhaustiva de todos los puertos y servicios, cuáles son los agujeros detectados y apuntando su solución. Muestra otras cualidades, como su permanente actualización para detectar nuevos agujeros de seguridad gracias a la fácil incorporación de nuevos plugins (para su fácil creación se ha creado un lenguaje específico denominado NASL). Características: ➢ Cada prueba de seguridad es escrita como un plug-in(conector externo). De esta forma, se puede agregar fácilmente pruebas propias sin tener que leer el código del motor de Nessues. ➢ El scanner de seguridad de Nessus incluye NASL (Nessues Attack Scripting Language), un lenguaje diseñado para escribir pruebas de seguridad. ➢ La base de datos de revisión de seguridad es actualizada diaramente. ➢ Se puede realizar la auditoría de toda la red desde una computadora personal. ➢ Dependiendo de la carga que pueda soportar el servidor de Nessue, se pueden examinar dos, diez o cuarenta computadoras anfitrión al mismo tiempo. ➢ Proporciona un reconocimiento inteligente de servicios. Nessues no cree que las conexiones de un host se respetarán los números de puertos asignados IANA. Esto significa que reconocerá un servidor FTP corriendo sobre un puerto no estándar (3030), o un servidor Web sobre el puerto 8080 ➢ En los reportes que presenta demás de mostrar que es lo que está mal en la red, dirá como prevenirse de los crackers que explotan los hoyos de seguridad encontrados, además de los niveles de riesgo de cada problema encontrado.
SAINT. Security Administrator's Integrated Network Tool. (FR-U)	SAINT es una versión actualizada de SATAN, es indispensable para revisar las vulnerabilidades del sistema.	En su modo simple reúne toda la información posible de host o redes remotas para examinar los servicios de redes, tales como finger, NFS, MIS ftp, tftp, rexd, statd entre otros. La información reunida incluye la presencia de varios servicios de información de redes, así como fallas de seguridad (instalación incorrecta de la configuración de red, conocidos como hoyos), políticas pobres. El potencial del SAINT se adquiere cuando se usa en su modo exploratorio. Basado sobre la colección de datos inicial y un conjunto de reglas configurable examinará las avenidas de confianza y dependencia, repitiendo estos procesos en computadoras anfitrión (hosts) secundarias. Esto permite analizar, además del host y la propia red, implicaciones reales inherentes a la red permitiendo tomar decisiones acerca de los niveles de seguridad del sistema involucrado. SAINT tiene un programa de adquisición de objetivos que usa fping para determinar si un host o conjunto de ellos en una red están presentes. Cuando un host está detrás de un firewall, tcp_scan es usado para probar puertos comunes y examinar que está presente. Después, pasa la lista de objetivos a una máquina que maneja una colección de datos , la cual regresa un diagnóstico. Cada host es examinado para ver si ha sido visto antes y si no es así, una lista de exámenes es ejecutada en él. Las pruebas emiten un registro de datos que tienen el nombre del host, la prueba ejecutada y cualquier resultado encontrado de la prueba; estos datos son salvados en archivos para análisis. La interfaz de usuario para leer los resultados es a través de HTML.

Tabla 4.22. Software de seguridad

Software	Objetivo	Descripción y características
Retina (Com-W)	Scanner de seguridad de redes y monitor que ayuda a descubrir y arreglar las vulnerabilidades de seguridad en sistema de Internet, Intranet o Extranet.	➢ Incluye fáciles herramientas de reporte de navegación para ayudar a decidir las prioridades, dando un control total sobre la auditoría la seguridad de la red y los getways de la red interna. ➢ Retina incluye la tecnología "Fix-It", una característica que permite automáticamene corregir asuntos de seguridad del sistema incluye establecer registros, permisos de archivos y otros. ➢ Incluye un módulo de inteligencia artificial el cual lo hace más inteligente acerca de localizar exactamente las vulnerabilidades. La lógica de la inteligencia artificial esta construida para manejar escenarios como "que tal si" y "en caso de que" ➢ Trabaja con módulos plug-in que pueden ser escritos en cualquier lenguaje de programación. eEye(Digital Security Team support) pronto será liberado como una API completa para el motor de reportes de Retina. ➢ Esto significa que se podrá escribir la rutinas de pruebas para una red especifica e incorporarlas en procedimientos de auditoría, incluyendo las propias alertas y acciones para ser consideradas al corregir el problema. Retina incluye módulos de auditoría para los siguientes sistemas y servicios: NetBIOS, HTTP, CGI, and WinCGI, ISAPI y ASP, FTP, DNS, POP, SMTP y LDAP TCP/IP y UDP, registros, servicios, cuentas de usuarios, vulnerabilidades de contraseñas, servidores de Bases de Datos SQL, firewalls, ruteadores, y servidores proxy.
ISS. Internet Security Scanner (Fr-U)	Es una herramienta de dominio público que verifica una serie de servicios para comprobar el nivel de seguridad que tiene un sistema Unix en búsqueda de ciertos huecos de seguridad.	ISS es capaz de verificar una dirección IP o un rango de direcciones IP (en este caso se indican dos direcciones IP e ISS verificará todas las máquinas dentro de ese rango). El programa viene acompañado de dos utilidades que son ypx y strobe, la primera nos permite la transferencia de mapas NIS a través de la red y la segunda verifica y describe todos los puertos TCP que tiene la máquina que verificamos. Con la primera herramienta es posible la transferencia de los archivos de "password" en aquellas máquinas que hayan sido configuradas como servidores de NIS.
SHTTP. Secure Hipertext Transfer Protocol. (Fr-U-W)	Aporta codificación y seguridad a nivel aplicación sobre comunicaciones ordinarias basadas en conexiones.	Incorpora codificación con llave pública de RSA. Además, soporta sistemas de seguridad, con uso de contraseñas basados en Kerberos. El cliente y el servidor se comunican en una sesión ordinaria y negocian sus requerimientos de seguridad. SHTTP aporta las siguientes características: ➢ Autentifica a clientes y a servidores ➢ Verifica revocaciones de certificados de servidores ➢ Soporta encadenamiento y jerarquías de certificadores ➢ Soporta firmas digitales que dan fe a la autenticidad de un mensaje ➢ Permite que una aplicación negocie el nivel de seguridad ➢ Brinda comunicación protegida por medio de Firewalls Proporciona codificación, autentificación, integridad de mensaje y no repudio.
SET (Secure Electronic Transactions) (Com-U-W)	Este protocolo esta especialmente diseñado para asegurar las transacciones por Internet que se pagan con tarjeta de crédito.	Es un conjunto de especificaciones desarrolladas por VISA y MasterCard, con el apoyo y asistencia de GTE, IBM, Microsoft, Netscape, SAIC, Terisa y Verisign, que permitirán el desarrollo del comercio electrónico en Internet y otras redes públicas, de forma segura para todos los participantes: usuario final, comerciante, entidades financieras, administradoras de tarjetas y propietarios de marcas de tarjetas. SET requiere un certificado digital en cada paso de autentificación y usa dos pares de llaves, una para el cifrado del sobre digital y otra para la firma, (SSL solo usa un par de llaves), actualmente SET usa la función hash SHA-1, DES y RSA de 1024 bits, estos parámetros fueron tomados para ser compatible con los certificados existentes, y se piensa que soporte también curvas elípticas en la próxima versión de SET.

Tabla 4. 22. Software de seguridad

Software	Objetivo	Descripción y características
Digicash (Com-U-W)	Este sistema esta basado en el sistema criptográfico de firmas digitales.	Es una compañia que implementa un sistema de pago llamado dinero digital, este trata de emular lo que es el dinero en efectivo y sus cualidades, es negociable, puede entregarse a cualquier persona, no se necesita identificarse quien lo emite, es almacenable, no requiere autorización para usarlo. Se requiere tener una cuenta con la compañía e instalar un software en la computadora cliente. Lo que hace el software es producir números del formato de la serie del dinero digital, la compañía firma estos con su firma digital y los respalda como si fuera dinero en efectivo, en general el cliente debe tener una cuenta bancaria para respaldar sus requerimientos de dinero digital. El dinero digital queda almacenado en la computadora cliente, la compañía tiene un sistema de verificación de las series para evitar la duplicación del dinero digital por parte del cliente. Se han seguido haciendo intentos por encontrar algún sistema que sea estandarizado ya que es muy difícil para los bancos trabajar con muchos sistemas a la vez.
Hardlock (Com-U-W)	Hardware que previene que los softwares y/o datos sean copiados o utilizados ilegalmente	Fue diseñado para prevenir ataques de hackers y emuladores de llaves. Tiene diversas aplicaciones y conexiones (puerto de impresión, ranura PCMCIA, ranura ISA, USB y puerto serie). El sistema Hardclock trabaja a través de una combinación de software y hardware, el cual esta compuesto por tres componentes básico: 1. El Hardclock 2. La placa Cripto-Programadora 3. El software Hardclock Bistro. Para utilizarse el Hardclock se debe utilizar la Placa Cripto-Programadora con un algoritmo exclusivo. Una vez programado, es incluido en cada programa de software. El programa buscara y verificará, con la periodicidad que se le haya indicado, la presencia del Hardclock. Si éste no se encuentra, la aplicación no correrá.

Tabla 4. 22. Software de seguridad

Es importante resaltar que los sistemas llamados scanners de seguridad se encargan principalmente de revisar que no existan agujeros en el sistema de su propia red. Tomemos en cuenta que no realiza la protección directamente en los datos sino explora la red (o el host) en busca de puertos y servicios para descubrir agujeros antes de que los intrusos lo hagan y de esta forma prevenir ataques, sin embargo si un husmeador pasa desapercibido por el scanner entonces ¿la información que se maneje en esa red es vulnerable a él? Además, si estas herramientas fueran utilizadas con diferente propósito, por ejemplo si un atacante las empleará para conocer como está constituida una red, ésta corre el riesgo de ser controlada por esa persona. Es por ello, que existen sistemas que además de incluir este tipo de herramientas utilizan la criptografía como medio para proteger directamente los datos, siendo a su vez una manera más difícil para los hackers de obtener información. Aparte de éste método, también se consideran otros mecanismos (autentificación, firmas digitales, compendios, etc.) para desarrollar un software más robusto desde el punto de vista de la seguridad se debe de utilizar técnicas como las mencionadas en el capítulo 1 (Tabla 1.3), y de esta forma obtener mejores resultados.

En la tabla 4.23 se reúnen las características más sobresalientes de los principales métodos de seguridad vistos. ¿Cuál de los mecanismos de cifrado ofrecen mayor seguridad? El primer ataque a cualquiera de los métodos de cifrado es la fuerza bruta y dependiendo de la complejidad del algoritmo de cifrado va a ser el tiempo que se emplee para obtener el texto natural. Sin embargo, al irse creando algoritmos de cifrado con funciones matemáticas más complejas, la fuerza bruta no era la mejor solución ya que además de emplear bastante tiempo se requería forzosamente máquinas capaces de realizar cómputo de carga pesada. Debido a ello, empezaron a diseñarse otras técnicas de ataque.

Por ejemplo, uno de los ataques que se aplican en sistemas que emplean cifrado en bloque con cajas S (en el caso de DES), es el criptoanálisis diferencial. Por supuesto que implica tener conocimiento de cómo funciona el algoritmo de cifrado, obtener un bloque de la información a cifrar y su correspondiente cifrado con el propósito de poder realizar un análisis sobre el cambio que sufren los bits para obtener la llave. Otro ataque "más fácil" en los sistemas simétricos, y que los hacen más vulnerable en comparación con los sistemas de llave pública, es la captura de la llave en cualquier medio de comunicación. Para que una persona descifre los datos que ha recibido requiere de la única llave que esta en poder de la persona que cifró dichos datos, entonces es necesario que se la envíen. Uno de los medios que puede emplear es por teléfono, correo electrónico o en forma verbal, pero tal vez estuvo presente un husmeador que para su suerte pudo capturar la llave, dando el poder de descifrar ó afectar la integridad de los mensajes de los usuarios que empleen esa llave.

El problema de la transmisión de llaves se resolvió con la aparición de los sistemas de llave pública. La ventaja de este método es que se utilizan dos llaves una para cifrar (pública) y otra para descifrar (privada). La llave pública puede estar al alcance de cualquier persona que desee enviar un mensaje cifrado a la persona que conocer la llave privada y que es la única en tenerla, evitándose el problema de exponer la llave privada al conocimiento de personas ajenas. Esto no quiere decir que los sistemas de llave pública sean invulnerables, existen técnicas de criptoanálisis que utilizan la factorización para encontrar la llave privada, pero el tiempo que tarden en descubrirla dependerá de que tan grande sea su longitud. Sin embargo, aún mediante el uso de técnicas de cifrado, no se puede garantizar que llegue a su destino en forma integra. Una vez que el mensaje fue cifrado es transmitido en paquetes a su destino, pero durante el transcurso de su camino pudiera existir una tercera persona ajena que capaz de capturar algunos paquetes de esa conexión, y como el cifrado se hizo a nivel aplicación entonces la información de la dirección destino y el canal por el cual existe la comunicación están al descubierto, entonces el husmeador le será fácil alterar los paquetes y enviarlos al receptor que esta esperando el mensaje. Claro que éste al descifrar el mensaje no obtendrá el original.

Método	Mecanismo de seguridad	Longitud de la llave	Características	Seguridad
DES	Cifrado simétrico	56 bits	Cifra en bloques de 64 bits. Se basa principalmente en transposiciones y sustituciones mediante operaciones booleanas.	Ha sido descifrado mediante el criptoanálisis diferencial
IDEA	Cifrado simétrico	128 bits	La llave sirve para generar 52 subllaves para cifrar 4 bloques de 16 bits utilizando operaciones OR EXCLUSIVO, suma módulo 2^{16} y multiplicación módulo $2^{16}+1$.	Este algoritmo se diseñó para no ser tan frágil ante el criptoanálisis diferencial
RSA	Cifrado de llave pública y Firma digital	Hasta 465 bits	Multiplicación y exponenciación modular	Ataques mediante la factorización
ECC	Cifrado de llave pública y Firma digital	Hasta 2048 bits	Operaciones de sumas sobre puntos de una curva elíptica	Ataques por medio de cálculos logaritmos de curva elíptica, y también por medio de la factorización.
MD5	Compendio de mensajes.	No usa llave	Función de dispersión unidireccional, toma una parte arbitraria del texto y a partir de ella calcula una cadena de bits de longitud fija. Procesa datos de entrada en bloques de 512.	Han existido ataques contra MD5, uno de ellos consiste en generar una colisión en la función de compresión.

Tabla 4.23. Algoritmos, protocolos e implementaciones de seguridad.

Método	Mecanismo de seguridad	Longitud de la llave	Características	Seguridad
SHA	Compendio de mensajes.	No usa llave	Procesa datos de entrada en bloques de 512.	Es un estándar del gobierno, todavía no se sabe si este algoritmo ha sido vulnerable a un ataque.
SSL	Autentificación, compendio de mensajes y cifrado	Depende del algoritmo de cifrado	Para el cifrado utiliza algoritmos, tales como IDEA, RC2-40, DES-40 DES, 3DES, Fortezza, RC4-40 y RC4-128. En el intercambio de llaves utiliza RSA ó Diffie-Hellman Y en la autentificación utiliza certificados X.509v3	Su vulnerabilidad depende de los algoritmos de cifrado que emplea. Sin embargo, este protocolo ofrece: 1. Integridad. La garantía de que los mensajes que enviamos o recibimos no han sido modificados. 2.Confidencialidad. Nadie sin autorización puede leer la información transmitida, y garantiza que efectivamente la reciba quien debe recibirla.
Kerberos V.5	Autentificación y encriptamiento	Depende del algoritmo de cifrado	El objetivo de Kerberos es permitir que un cliente mediante el nombre de un usuario particular pruebe su identidad a un servicio, o al correspondiente servidor de aplicaciones, sin necesidad de enviar datos por la red que podrían facilitar el que un atacante suplantase posteriormente al usuario. Emplea como algoritmo de cifrado DES y el manejo de billetes (ó certificados) para autentificar a usuarios	Kerberos requiere de la existencia de una tercera entidad de confianza que actúe como centro de distribución de llaves (KDC, Key Distribution Center). El KDC está constituido de dos elementos: 1. AS(Servidor de autentificación). 2. TGS (Servidor de emisión de billetes) Su seguridad radica en los tres intercambios que se realizan con estos elementos: ➢ El intercambio entre el cliente y el AS ➢ El intercambio entre el cliente y el TGS . ➢ Autentificación entre el cliente y el servidor de la aplicación.
Firewalls	Filtrado de paquetes Gateways a nivel aplicación y circuito		En el filtrado de paquetes el ruteador se encarga de examinar los paquetes, si alguno de ellos no satisface el criterio es descartado.	Uno de los ataques utilizados con más frecuencia para saltarse la protección establecida por un firewall sobre filtrado de paquetes consiste en generar paquetes IP con direcciones falsas.
			Los gateways a nivel aplicación se encargan de examinar el contenido de cada paquete que viaja a través de la conexión solicitada por un usuario a través de un servicio. Se ayuda mediante la implementación de un servidor proxy.	Las pasarelas a nivel de aplicación proporcionan una puerta de acceso para cualquier servicio (FTP, Telnet, Correo Electrónico).
			Los gateway a nivel circuito se encargan de determinar cuales conexiones serán permitida.	Las pasarelas a nivel de circuito trabajando con los servidores proxy son utilizadas para la acreditación, es decir, comprobaciones sobre máquina fuente, máquina destino, puerto a utilizar. Si la acreditación es positiva, se establece la conexión.

Tabla 4.23. Algoritmos, protocolos e implementaciones de seguridad

Para resolver este problema se lleva a cabo el mecanismo de autentificación mediante el uso de firmas digitales, incluyendo un compendio del mensaje calculado por funciones de dispersión ó Hash (por ejemplo, MD5 ó SHA-1). Éstas realmente lo que hacen es comprimir el mensaje de modo tal que producen un "resumen", el cual será comprobado por el receptor con su firma digital, para asegurar que se mantiene la integridad del mensaje y que verifica que el emisor realmente es la persona con la que esta llevando la comunicación.

Además de realizar la protección de datos en forma directa existen otras técnicas que tratan de evitar que personas ajenas acedan a ellos utilizando técnicas de control de acceso a la información. En ellas se seleccionan o se filtran a los usuarios para hacer uso de los recursos informáticos. Para llevar a cabo tal selección se instalan filtros en la red

mediante ruteadores o firewalls para controlar los accesos desde máquinas remotas. Sin embargo, la seguridad que ofrezcan dependerá de la configuración y mantenimiento que se hagan en ellos.

En los niveles del Modelo de Refencia OSI se pueden implantar técnicas de seguridad que garanticen la integra transferencia de datos. Por ejemplo, en la capa de enlace de datos, los paquetes de una línea punto a punto pueden codificarse al ser enviados desde una máquina y descodificarse cuando lleguen a otra. Sin embargo, esta solución tendría la desventaja de que al pasar los paquetes por los ruteadores, estos se descifrarían quedando vulnerables a ataques dentro del ruteador.

En la capa de red, pueden instalarse muros de seguridad (firewalls) que consisten de ruteadores y/o pasarelas (getways) cuya función es la de filtrar los paquetes que entran o salen de la red, así si los paquetes satisfacen algún criterio se reenvían normalmente. Los que fallan la prueba se descartan. Actualmente, se esta implementando la ***versión seis del protocolo IP***, en el cual se considera el cifrado de la carga útil.

En la capa de transporte, pueden cifrarse conexiones enteras extremo a extremo, por ejemplo, el protocolo SSH. Sin embargo, la mayoría del software de seguridad se hace a nivel aplicación, debido a su fácil diseño e implementación (en comparación con los anteriores), tal vez no ofrecen la seguridad que otros que se han implementado en capas más bajas pero todo depende de que tan valiosa es la información que se maneje. En algunos productos comerciales la seguridad que ofrecen sobre las transacciones de datos es solamente el encriptamiento, otros aumentan el nivel de seguridad al agregar métodos de autenticidad y firmas digitales.

Capítulo 5

Aplicación de Seguridad EncripData

En este capítulo se mostrará el diseño de un programa con propiedades de seguridad con el propósito de consolidar los aspectos teóricos vistos en los capítulos anteriores de este trabajo de tesis. Esta herramienta se denomina EncripData y fue desarrollada para realizar una comunicación segura entre dos o más usuarios en tiempo real, de tal forma que la información que manejen por medio de ella no sea perceptible por husmeadores.

5.1. Esquema general.

Por medio de Internet, usuarios de diferentes partes del mundo pueden comunicarse sin importan la distancia a la que se encuentren. Para una persona que está en México es igual de fácil comunicarse con alguien ubicado en la casa de al lado o en Japón (aunque el tiempo de respuesta es mayor). Hay varios sistemas de comunicación a través de Internet, el más común es el correo electrónico, que sirve para recibir y enviar mensajes escritos almacenados en un servidor de correo. Otro método es la conferencia telefónica, en la cual dos usuarios pueden tener una conversación similar a una charla por teléfono (para hablar y escuchar, los usuarios usan el micrófono). La videoconferencia, por su parte, permite ver en la pantalla al expositor. Los problemas de esta herramienta se ven en su alto costo (se requiere de cámaras de video y la línea telefónica adecuada que proporcione este servicio).

Otro sistema es el conocido como cuartos de conversación o Internet Relay Chat (IRC), el cual, tiene ventajas sobre los anteriores: no requiere multimedia y, a diferencia del correo electrónico, la comunicación es inmediata. Cada uno de los participantes ve en la pantalla de su computadora lo que él mismo escribe y lo que responden los demás; en una sesión pueden intervenir varias personas.

El IRC usa un programa cliente que sirve de interfaz al usuario y que se conecta a un servidor IRC. Cada uno de estos servidores están conectados a los servidores más cercanos, formando una telaraña para que los usuarios de diferentes lugares puedan conversar. Para poner orden, IRC mantiene un número de diferentes canales. Cuando un usuario se conecta al IRC, primero elige a que canal se incorpora para después poder hablar con otra persona en él. Si quiere también puede incorporarse a más de un canal a la vez.

Hay dos tipos de canales permitidos en el IRC. Uno es el de los canales distribuidos los cuales son conocidos por todos los servidores que están conectados a la red. Mientras exista un canal de este tipo, cualquier cliente puede referirse a él usando su nombre. Entonces los usuarios de Internet, aunque no estén en un canal, puede saber lo que otras personas están conversando.

El otro tipo de canal es el llamado privado creado por dos clientes que no quieren que otros usuarios intervengan en su conversación. En realidad, este canal sólo existe en un servidor y únicamente los clientes conectados a él tienen permiso de entrar al canal.

El inconveniente de este tipo de comunicación es que si usuarios (clientes) decidieran llevar a cabo una conversación importante de negocios y si por casualidad una persona ajena a ellos estuviera interceptando sus mensajes, le sería muy fácil interpretarlos ya que tales mensajes son simplemente datos codificados en formato ASCII. Y entonces, podría vender esa información a la competencia, o bien alterarla.

¿Cómo capturaría el husmeador los mensajes? Aunque la conversación se lleve a cabo en un canal privado, el husmeador sabe que en cualquier instante se llevará a cabo al menos una conversación en el servidor IRC, por lo que decidirá capturar los paquetes que lleguen y salgan del servidor mediante los analizadores de paquetes de red que se pueden obtener como ***freeware*** en Internet. Entre la información capturada encontraría el canal o canales donde se llevan a cabo las conversaciones, quienes son los clientes y los mensajes.

Como se observa, el anterior tipo de comunicación no ofrece a los clientes confidencialidad en su conversación. Una forma de lograrlo es ocultar los datos codificados en ASCII, además de eliminar la intervención del servidor, de tal forma que se podría "asegurar" que la comunicación únicamente sería realizada entre los usuarios interesados (una especie de canal privado).

El propósito de este trabajo de tesis es presentar un software que ofrezca el mecanismo de confidencialidad sobre la información manejada por usuarios que realicen transacciones de información inmediata. La aplicación propuesta se llama EncripData y emplea la criptografía de ocultar la información transmitida a personas ajenas a ella.

Debido a la actual comercialización de las redes Microsoft, su uso se ha vuelto muy popular en oficinas gubernamentales, empresas privadas, instituciones educacionales entre otras. Para ofrecer a estos usuarios un medio de comunicación basado en texto en tiempo real, el sistema EncripData se ha diseñado para trabajar sobre redes Microsoft. Para usarlo requiere ser instalado en una máquina que este conectada en red y con Windows (NT, 95 ó 98) cuyo subsistema de red debe estar configurado para trabajar sobre los protocolos NetBIOS y TCP/IP.

El funcionamiento general de EncripData se basa en el modelo cliente-servidor, es decir, involucra un proceso único (el servidor) formando el punto central donde se conecta los clientes (usuarios), para realizar la recepción de mensajes y posteriormente las actualizaciones a cada cliente que está en el grupo de conversación.

Antes de empezar a explicar la aplicación EncripData es conveniente presentar un bosquejo general de su funcionamiento y así mostrar cuál es el problema a resolver. En el siguiente ejemplo sólo se lleva a cabo la comunicación entre dos usuarios finales con el fin de hacer el proceso más sencillo de describir, sin embargo hay que considerar que la aplicación esta diseñada para trabajar con un grupo de usuarios.

La situación es la siguiente, existen dos máquinas con Windows 95 llamadas W95A y W95B mostradas en la figura 5.1; cada una de ellas está ejecutando la aplicación EncripData.

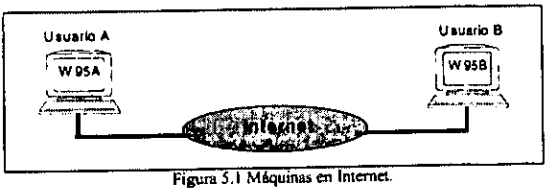


Figura 5.1 Máquinas en Internet.

El usuario A decide establecer una comunicación con el usuario B mediante EncripData, así que le envía un mensaje que anuncia una invitación para empezar una comunicación. Hay que hacer notar que EncripData utiliza un puerto no asignado (puerto X) para solicitar el servicio de comunicación con otra máquina que tenga la misma aplicación corriendo. Una vez que el usuario B recibió la solicitud, envía una respuesta (de aceptación o rechazo) al usuario A. Vea la figura 5.2.

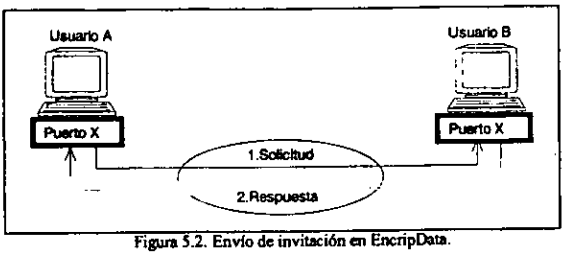


Figura 5.2. Envío de invitación en EncripData.

Como se observa en la figura siguiente, si la respuesta fue de aceptación, el usuario B transmite sus mensajes por otro puerto que el mismo sistema le asigna, en este caso es el puerto B.

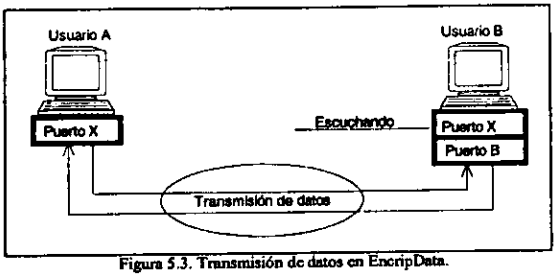


Figura 5.3. Transmisión de datos en EncripData.

Después de que ambas aplicaciones establecieron un canal de comunicación, los usuarios A y B podrán conversar textualmente. Es posible que el usuario B pueda escuchar simultáneamente otras solicitudes de invitación por el puerto X y si él acepta tales solicitudes podrá pertenecer a otros grupos de conversación independientes al que tiene con el usuario A; o bien , él puede hacer sus invitaciones y crear uno o más grupos de conversación. Ahora corresponde hablar acerca de la seguridad que ofrece la aplicación propuesta.

Los mensajes que se transfieren en el canal pueden ser capturados por cualquier husmeador que esté en la red y por consiguiente saber su contenido. Una de las formas para evitar que los mensajes sean entendibles por otras personas ajenas es por medio de la criptografía. EncripData utiliza este método de seguridad, específicamente emplea el

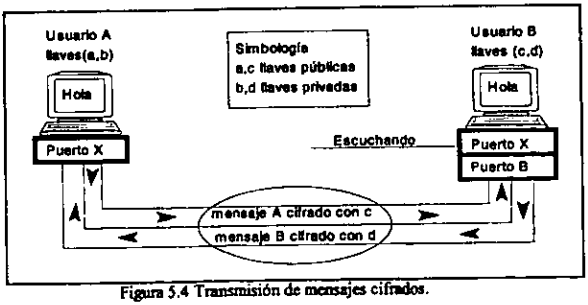


Figura 5.4 Transmisión de mensajes cifrados.

algoritmo RSA para encriptar los mensajes. Hay que tomar en cuenta que para llevar a cabo el cifrado de los datos se requiere de un par de llaves (pública y privada), entonces, cada usuario que utilice la aplicación deberá tener su propio par. Observe la figura 5.4.

¿Porqué se seleccionó RSA como el algoritmo de cifrado empleado en EncripData? Se seleccionó un algoritmo de llave pública porque como vimos en el capítulo 4, una ventaja que ofrece este sistema es que resuelve el problema de la transmisión de llaves haciendo más "confiable" su intercambio. Además, de que la longitud de las llaves pueden ser más grandes que la de los sistemas simétricos, y como anteriormente se expuso el tiempo que tarden en descubrirla dependerá de esta longitud.

5.2. EncripData.

El desarrollo de la aplicación EncripData está dividido en las siguientes partes: análisis, diseño, construcción, y pruebas. A continuación se presentan cada una de ellas en el orden que fueron señaladas.

5.2.1. Análisis.

Partamos de la idea de que el sistema debe ser capaz de trasmitir un mensaje tecleado por un usuario a otro que se encuentra en una red local. Primero, se requiere que el usuario emisor proporcione la dirección de la máquina destino (dirección IP o nombre de máquina) y el mensaje a enviar. Después la aplicación se encargará de transmitir estos datos a la correspondiente máquina, la cual mediante la misma aplicación recibirá los datos para después desplegarlos al usuario receptor. Este proceso se muestra en la siguiente figura.

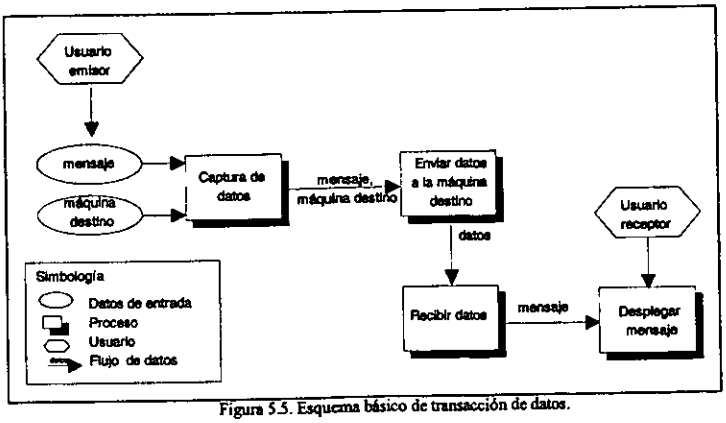


Figura 5.5. Esquema básico de transacción de datos.

Sin embargo, la conversación en tiempo real en un grupo de usuarios no consiste únicamente del proceso anterior, hay que considerar otros aspectos. Si un usuario desea establecer una conversación con sus compañeros primero debe

preguntarles si están de acuerdo en llevarla a cabo. Para esto se ha establecido un protocolo de inicio de conversación que se muestra en la figura 5.6, y donde el usuario que llama (iniciador del grupo de conversación) envía una invitación a los usuarios con los que se quiere comunicar (posibles miembros del grupo de conversación). Si cada uno de ellos acepta la invitación, regresarán una respuesta afirmativa al iniciador pero si uno la rechaza su respuesta será negativa (Figura 5.6(b)). En el caso de que todos contesten en forma negativa no se creará el grupo de conversación.

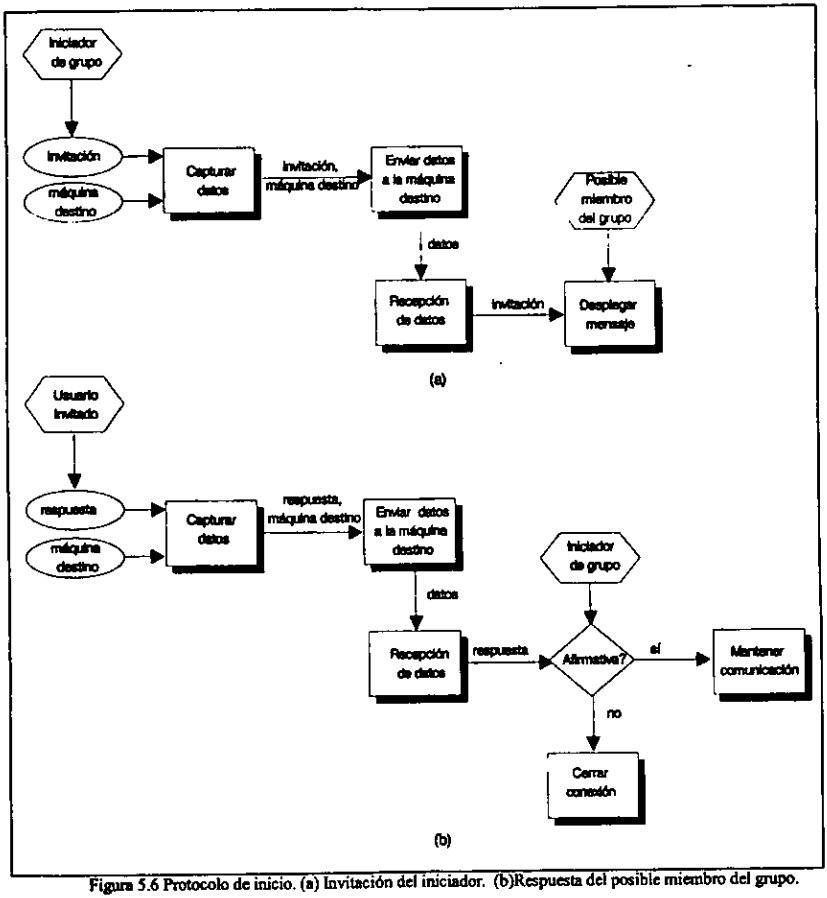


Figura 5.6 Protocolo de inicio. (a) Invitación del iniciador. (b)Respuesta del posible miembro del grupo.

Después de que el iniciador y los miembros del grupo de conversación han establecido una conversación, ¿Cómo se distribuyen los mensajes? ¿Todos los usuarios distribuyen los mensajes para todos? Para contestar a estas preguntas se debe considerar que el primer interesado en transmitir un mensaje a otros usuarios es el iniciador de grupo, además él es quien selecciona a los integrantes de la conversación. Entonces, el iniciador del grupo es quien va llevar el control de la conversación. En realidad, la aplicación del iniciador de grupo se convierte en servidor cuando al

menos un posible miembro del grupo acepta la invitación, y la aplicación de éste último se convierte en cliente. Después, el servidor se encargará de distribuir los mensajes que los miembros del grupo le envíen, de esta forma actualiza la conversación a todos (Figura 5.7). Cuando el iniciador del grupo decide dar por terminada la comunicación (como él es quien tiene el control), desconecta a todos los miembros del grupo de conversación.

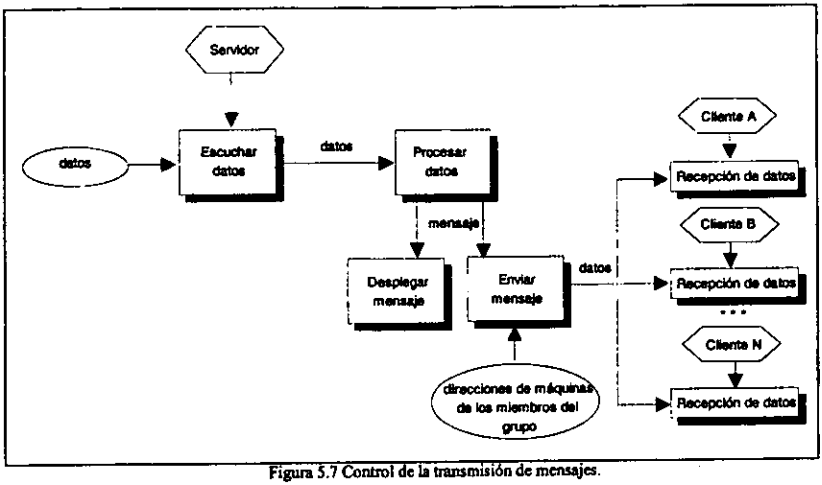


Figura 5.7 Control de la transmisión de mensajes.

La siguiente tabla contiene el significado de algunos términos utilizados en los párrafos anteriores:

Término	Significado
Iniciador de grupo	Usuario que desea abrir un grupo de conversación
Usuario invitado o posible miembro de grupo	Usuario que recibe una invitación para formar parte de un grupo de conversación.
Miembro del grupo de conversación	Usuario que ya es integrante de un grupo de conversación.
Servidor	Este término lo recibe la aplicación EncripData del iniciador de grupo cuando se crea el grupo de conversación
Cliente	Este término lo recibe la aplicación EncripData de un miembro del grupo.

Tabla 5.1. Términos.

Hasta el momento sólo se ha analizado la parte de comunicación, falta ver cómo y cuáles métodos de seguridad serán implementados en la aplicación EncripData.

En el anterior capítulo se mostró que el método de seguridad de la información más utilizado para realizar transacciones de datos seguras es la criptografía. Por ello, se ha pensado en utilizar uno de los algoritmos más populares para realizar el cifrado de los datos conocido como RSA.

Para implementar el proceso de seguridad en la aplicación debemos hacer algunas modificaciones al modelo del protocolo de inicio. Ahora el iniciador del grupo proporcionará como datos las direcciones de la máquina destinos, la invitación y su llave pública (Figura 5.8(a)); de tal forma que, si los posibles miembros del grupo de conversación aceptan la invitación cifrarán la respuesta afirmativa con la llave pública del iniciador, de lo contrario enviarán una respuesta negativa sin cifrar.

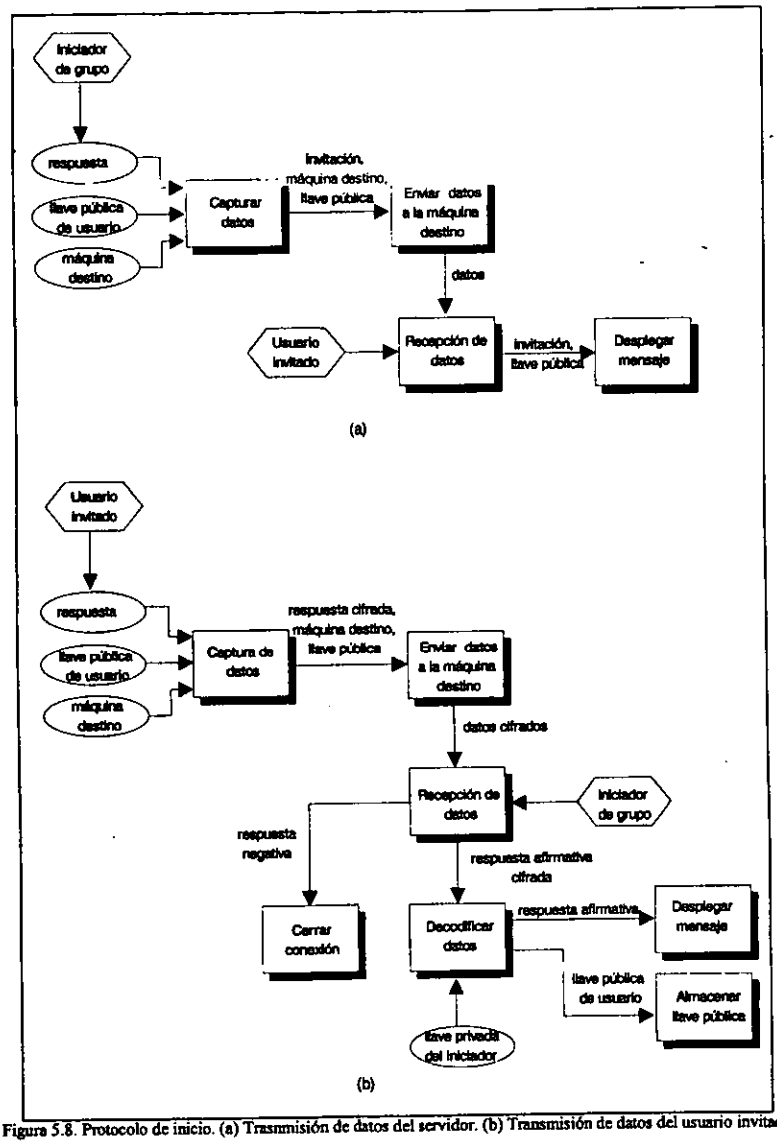


Figura 5.8. Protocolo de inicio. (a) Transmisión de datos del servidor. (b) Transmisión de datos del usuario invitado.

En la figura 5.8(b) se observa que en el caso de que los miembros del grupo acepten la invitación les corresponde almacenar la llave pública del iniciador para cifrar los futuros mensajes a enviar.

Entonces, cada vez que la aplicación servidor reciba datos los decodificará con la llave privada del iniciador de grupo y antes de que envíe un mensaje deberá cifrarlo para cada miembro del grupo de conversación con la respectiva llave pública. Observe la figura 5.9(a).

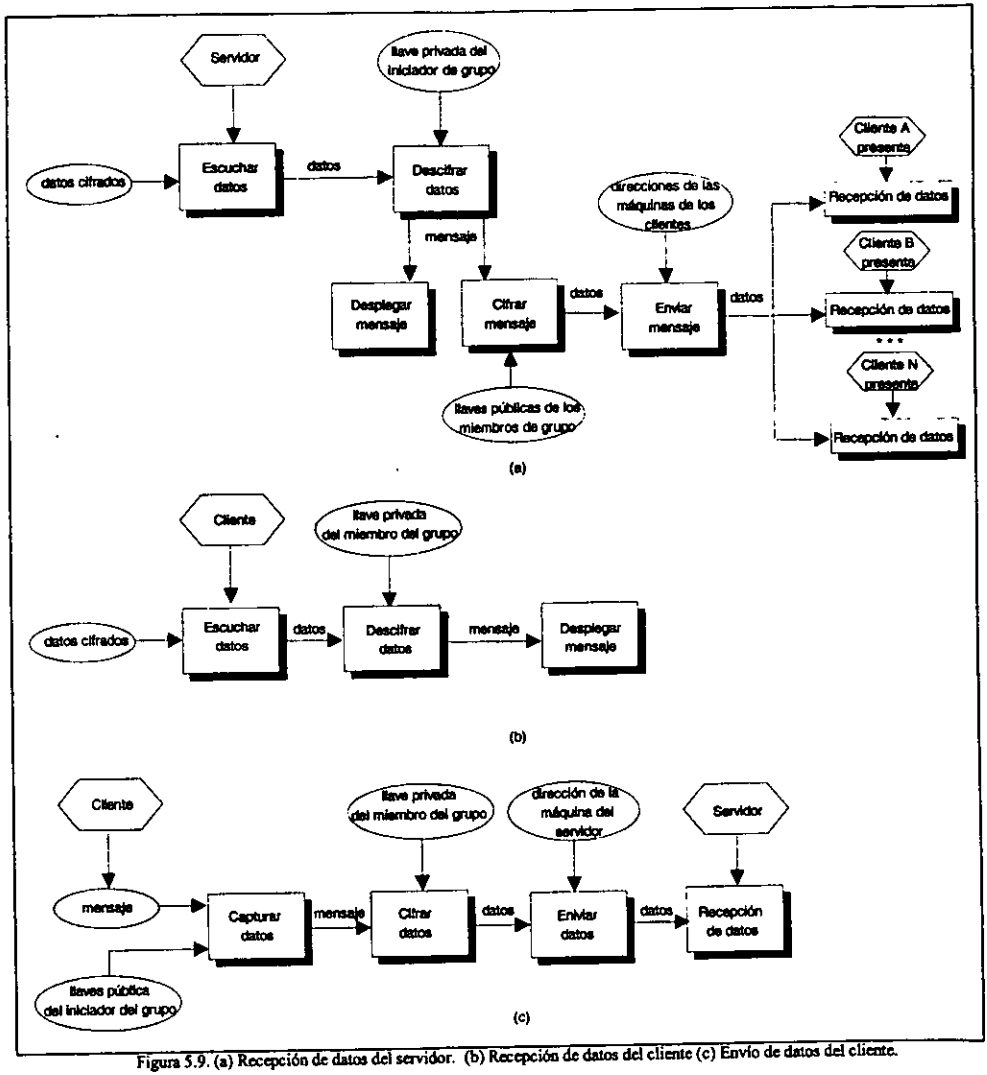


Figura 5.9. (a) Recepción de datos del servidor. (b) Recepción de datos del cliente (c) Envío de datos del cliente.

En la figura 5.9(b) podemos observar que el cliente usa su llave privada para descifrar el mensaje. Y en la figura 5.9(c), el cliente utiliza la llave pública del iniciador de grupo para cifrar el mensaje.

5.2.2. Diseño y construcción.

Después de plantear el problema veamos ahora como resolverlo. Antiguamente los programas eran más sencillos y no requerían demasiadas líneas de código, pero ahora pueden ser muy complejos y tener una gran cantidad de líneas de código. Cuando se llega a este punto la programación tradicional puede complicarse, por lo que hace falta una manera eficaz de organizar los programas. La programación orientada a objetos descompone los problemas en subgrupos relacionados de manera que la complejidad del programa se reduzca. Es por ello que se pensó en desarrollar la aplicación utilizando un lenguaje de programación orientada a objetos. Para crear la aplicación EncripData se ha utilizado como herramienta de programación Visual C++. Uno de los beneficios de emplear Visual C++ es el de diseñar y crear las interfaces de usuario final en poco tiempo, permitiendo así concentrarse más en la parte de la programación orientada a objetos. Sin embargo, para empezar a escribir el código es necesario tener el diseño de la solución del problema a resolver.

Al realizar un diseño que involucre objetos es importante saber en que consiste cada uno de ellos. Los objetos son entidades lógicas que contiene datos y un código que manipula esos datos. Cuando se escribe un programa en

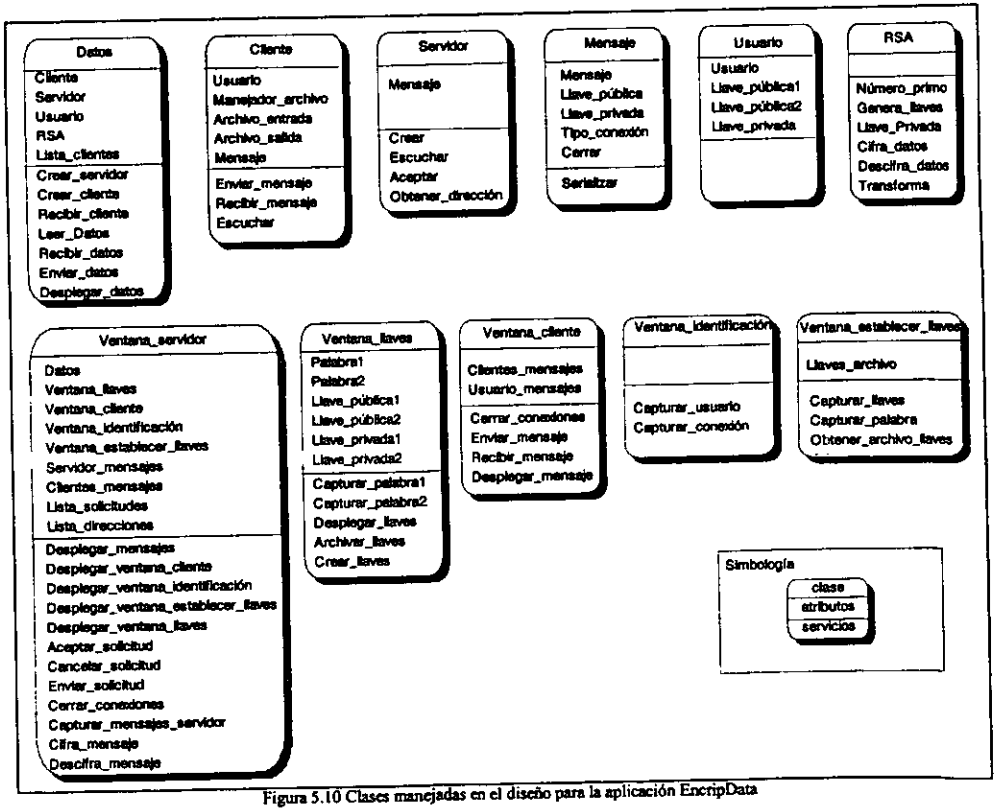


Figura 5.10 Clases manejadas en el diseño para la aplicación EncripData

lenguaje orientado a objetos, no se definen los objetos verdaderos; se definen las clases de esos objetos. Entonces, para el diseño de EncripData se han creado las clases mostradas en la figura 5.10.

Las clases Servidor y Cliente son empleadas para crear sockets. Un socket es la representación abstracta del extremo de la comunicación en red y al crearlo se le asocia un puerto identificado por un número. Cuando se crea un socket a partir de la clase Servidor se le asocia un número de puerto (asignado por el programador), que es registrado por el sistema operativo. Entonces, cualquier dato que llegue a ese puerto es enviado a la aplicación en donde se creó el socket asociado. En el caso de que se cree un socket Cliente, la misma aplicación le asignará el número de puerto. Los primeros 1024 puertos están reservados para servicios y mayores ese número hasta los 65,536 son destinados para conexiones clientes.

El puerto que se le ha asignado a la aplicación EncripData para que actúe como servidor es el 700, pudo seleccionarse cualquier número entre 200 al 1024. Los primeros 200 ya son servicios definidos (ver tabla B.1 del apéndice B).

La clase Mensaje es utilizada para crear un objeto destinado a encapsular o desencapsular los datos (enviados o recibidos respectivamente), tanto del servidor como del cliente. Este proceso lo realiza la función *Serializar*, si los datos son de salida los almacena en un archivo de salida conservando sus características. En cambio, si llega un archivo los datos son extraídos y almacenados en el tipo de variables correspondientes. Vea la figura 5.11.

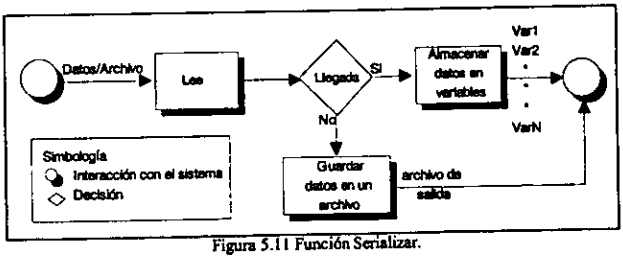


Figura 5.11 Función Serializar.

Cuando el *Servidor* lea los datos enviados por el cliente tendrá que accesar al objeto *Mensaje* del correspondiente *Cliente*. Y cuando el *Servidor* envíe datos a los clientes, tendrá que usar el archivo de salida del objeto *Mensaje* y la dirección de cada *Cliente*. Los mensajes que se manejan en esta aplicación son un flujo continuo de caracteres limitados por el retorno de carro. La representación BNF[8] de un mensaje es la siguiente:

<mensaje> :: <nombre_usuario> : <datos> <retorno de carro>
<nombre_usuario>::(<dígitos>|<caracteres>)+
<datos>:: (<dígitos>|<caracteres>)*
<caracteres>::(<alfabeto>|<especial>|<puntuación>)*
<especial>::!|@|#|$|%|&|/|=|+|-|*|(|)|{|}|[|
<puntuación>:: ¬ ' | . | , | : | ; | ' | "
*alfabeto = las 29 letras del alfabeto castellano tanto mayúsculas como

Tabla 5.2.Represntación BNF de un mensaje.

[8] BNF(Backus Naur form), es una forma expresar las ideas para lenguajes de programación.

Existen algunos mensajes definidos por la aplicación utilizados en el protocolo de inicio y cuando se cierra una conexión. En la tabla 5.3 se muestran estos mensajes.

Mensaje	Formato
Invitación	<nombre_iniciador> : Incorporate al grupo de conversación
Aceptación	<nombre_invitado> : Estoy en el grupo de conversación
Rechazo	<nombre_invitado> : Por el momento estoy ocupado
Desconexión	<nombre_invitado> : conexión cerrada o <nombre_servidor>: Servidor desconectado

Tabla 5.3. Mensajes que se manejan en la aplicación

Las clases Ventana_servidor, Ventana_cliente, Ventana_llaves, Ventana_establecer_llaves y Ventana_identificación son utilizadas para crear las interfaces con las cuales ellas los usuarios finales podrán interactuar con la aplicación EncripData.

La clase Datos crea un objeto que sirve de enlace para la transmisión de datos entre las interfaces y los sockets. Otra clase importante es la de Usuario, mediante ella se crea un objeto que almacenará los datos del usuario (nombre y llaves de cifrado) y será manejado por la aplicación en todo momento.

Y por último, la clase RSA que en realidad es una librería dinámica que actúa como un objeto durante la ejecución de la aplicación. Cuando se crea un archivo ejecutable de un programa, la librería se vincula en tiempo de ejecución, por lo que el archivo ejecutable y la librería deben distribuirse juntos.

En la siguiente tabla se describen las clases mostradas en la figura 5.9.

Objeto	Descripción
Datos	Esta clase crea un objeto que maneja los datos del servidor y de los clientes del servidor.
Cliente	Clase que crea un objeto para manipular los datos del cliente
Servidor	Clase que crea un objeto para manipular los datos del servidor
Mensaje	Clase que crea el objeto mensaje para dar un formato al mensaje del servidor o cliente.
Usuario	Clase que crea un objeto que almacena datos del usuario, ya sea iniciador o miembro del grupo.
RSA	Este es un objeto cuyas funciones se ocupan para generar llaves de cifrado, cifrar y descifrar datos.
Ventana_servidor	Clase que crea la interfaz del servidor
Ventana_cliente	Clase que crea la interfaz el cliente
Ventan_llaves	Clase que crea la interfaz para generar las llaves de cifrado
Ventana_identificación	Clase que crea la interfaz donde el usuario proporcionará su identificación durante la conversación.
Ventana_capturar_llaves	Clase que crea la interfaz para la captura de las llaves de cifrado del usuario.

Tabla 5.4. Clases de la aplicación EncripData.

El diseño de la aplicación será dividido en dos partes: comunicación e interfaz de EncripData. A su vez, la parte de comunicación se ha subdividido en los procesos servidor, cliente, identificación de usuario y llaves. Éstos se explicarán utilizando diagramas orientados a objetos para mostrar las relaciones de los objetos creados a partir de sus clases.

5.2.2.1.Comunicación.

Proceso Servidor

Cuando el programa sea ejecutado se desplegará la interfaz de la aplicación EncripData creada mediante la clase *Ventana_servidor*. Después, este objeto creará otro llamado *Datos* que a su vez dará origen al objeto *Servidor* encargado de escuchar las conexiones de los clientes. El proceso se muestra en la figura 5.12 (ver apéndice D, código D.5 y D.6).

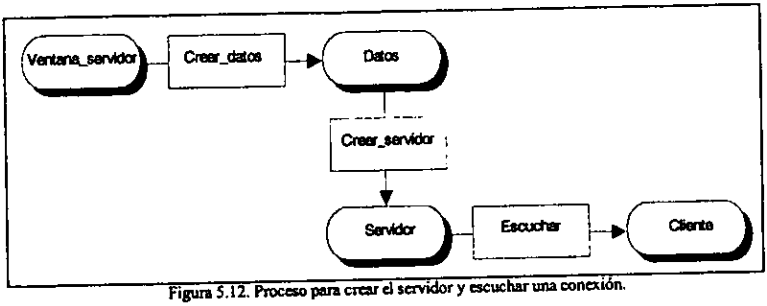


Figura 5.12. Proceso para crear el servidor y escuchar una conexión.

Después de crear el objeto servidor veamos que ocurre cuando un cliente decide conectarse con él. Como se observa en la figura 5.13, cuando el *Servidor* ha escuchado que alguien desea hacer una conexión, éste la acepta mediante la función miembro *Aceptar* y después llama a la función *Crear_cliente* del objeto *Datos* para crear un *Cliente*. El propósito de este objeto es para que el *Servidor* identifique la conexión del usuario. Entonces, por cada nueva conexión se tendrán un objeto *Cliente*.

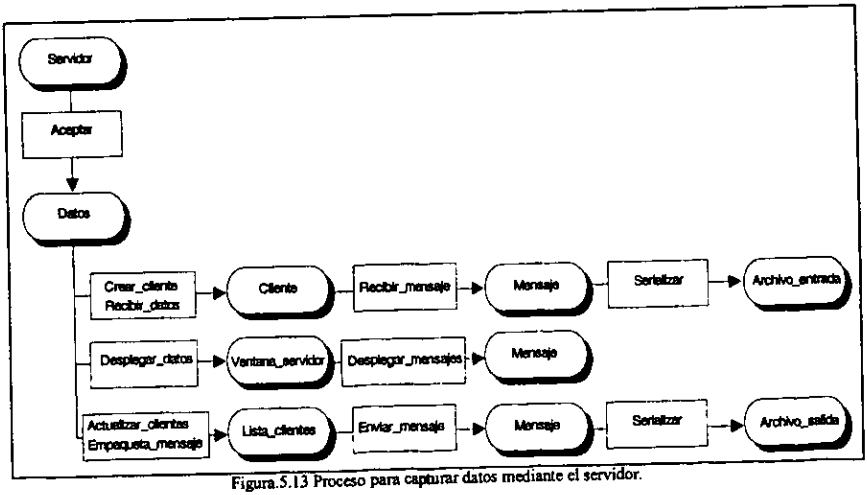


Figura.5.13 Proceso para capturar datos mediante el servidor.

Posteriormente, el *Cliente* llama a la función *Recibir_mensaje*, el cual crea un objeto *Mensaje* para obtener mediante la función *Serializar* los datos almacenados en el archivo de entrada.

Una vez que se han obtenido y almacenado los datos en el *Cliente*, el objeto *Datos* se encarga de definir el tipo de conexión del cliente (invitación o conversación) mediante la función *Recibir_cliente*. Después, con la función *Recibir_datos* almacena los datos para posteriormente ser desplegados por el objeto *Ventana_servidor* y enviados a los demás clientes por medio de las funciones *Actualizar_clientes*, *Empaqueta_mensaje* y *Enviar_mensaje*. Cabe señalar que la función *Enviar_mensaje* hace uso del objeto *Mensaje* para almacenar los datos en un archivo de salida. (Las anteriores funciones se encuentran en el apéndice D, en los códigos D.4, D.6, D.8 y D.10).

Ahora corresponde explicar el proceso cuando el iniciador del grupo desea enviar un mensaje y que se muestra en la figura 5.14. Primero, el usuario escribe el mensaje en la *Ventana_servidor* y mediante su función *Capturar_datos* lo trasmitirá al objeto *Datos*, el cual se encargará de actualizar a los clientes.

La función *Actualizar_clientes* le corresponde listar a todos los objetos *Cliente* y dar un formato al mensaje para después enviarlo mediante la función *Enviar_datos* a cada miembro del grupo de conversación. (Ver apéndice D, códigos D.4, D.8, D.10 y D.14)

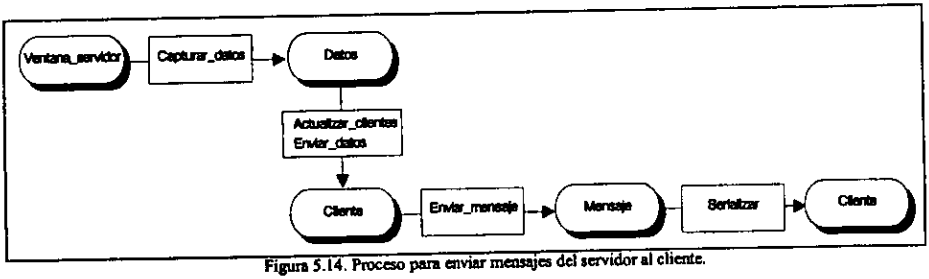


Figura 5.14. Proceso para enviar mensajes del servidor al cliente.

Proceso Cliente

El segundo proceso a explicar es la comunicación del cliente (figura 5.15). Este proceso se realiza a partir del objeto *Ventana_servidor*, y por medio de una de su función *Desplegar_ventana_cliente* se construye el objeto *Ventana_cliente* (esta es la interfaz con la que el usuario invitado interactúa). Después, este objeto crea el *Cliente* cuya función a ejecutar es enviar un mensaje de conexión al servidor y escuchar esa conexión. (códigos D.2, D.14 y D.16)

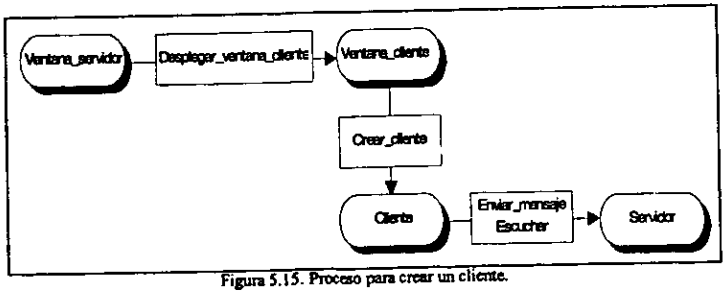


Figura 5.15. Proceso para crear un cliente.

Si observamos la figura 5.16, cuando el miembro del grupo decide enviar un mensaje, el objeto *Ventana_cliente* captura los datos tecleados y se los transmite al objeto *Cliente*, el cual llama a la función *Enviar_mensaje* que su a vez llama al objeto *Mensaje* para serializarlos y enviarlos por la conexión establecida. (códigos D.2, D.10 y D.16).

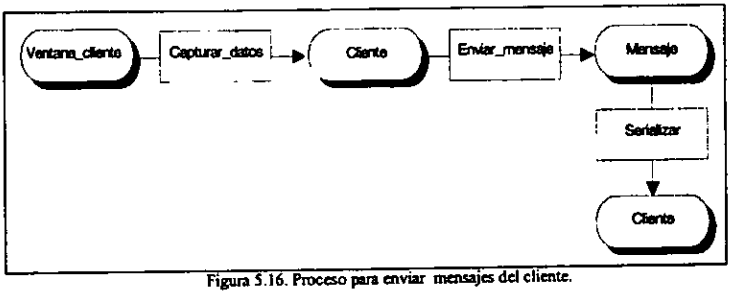


Figura 5.16. Proceso para enviar mensajes del cliente.

En la figura 5.17, se describe el proceso cuando el cliente escuche al servidor. El objeto cliente siempre está pendiente de la conexión que tiene con el servidor, si recibe datos que provengan de él, los serializa y los guarda en el objeto mensaje. Posteriormente, se llama al objeto *Ventana_cliente* para desplegar el mensaje del servidor. (Ver apéndice D, códigos D.2, D.10 y D.16)

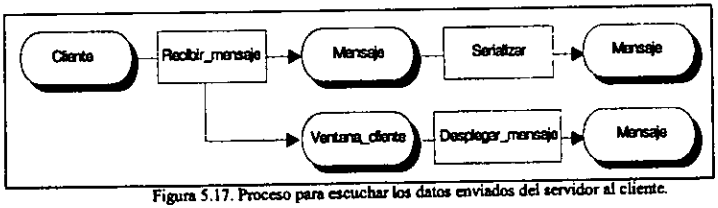


Figura 5.17. Proceso para escuchar los datos enviados del servidor al cliente.

Proceso Identificación de usuario

Corresponde explicar el proceso de identificación de usuario. Solamente el objeto *Ventana_servidor* puede crear el objeto *Ventana_ identificación* (ver apéndice D, códigos D.12, D.14 y D.20), el cual va a capturar el nombre de usuario y establecer si la conexión es segura o no, esto depende de lo que teclee el usuario final. Los datos son guardados en el objeto *Usuario* para su uso posterior, ya sea con el cliente o el servidor. Este proceso se observa en la figura 5.18.

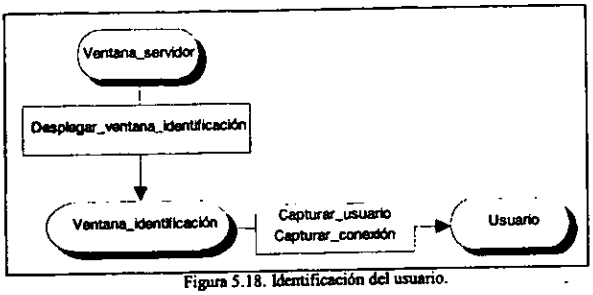


Figura 5.18. Identificación del usuario.

En el caso de que el usuario haya especificada la conexión como segura implica otro proceso que se muestra en la figura 5.19. Una vez que son capturados los datos del usuario, el objeto *Ventana_servidor* crea la *Ventana_establecer_llaves* (código D.17 y D.18 del apéndice D), encargada de capturar las llaves de cifrado del usuario para almacenarlas en el objeto *Usuario*

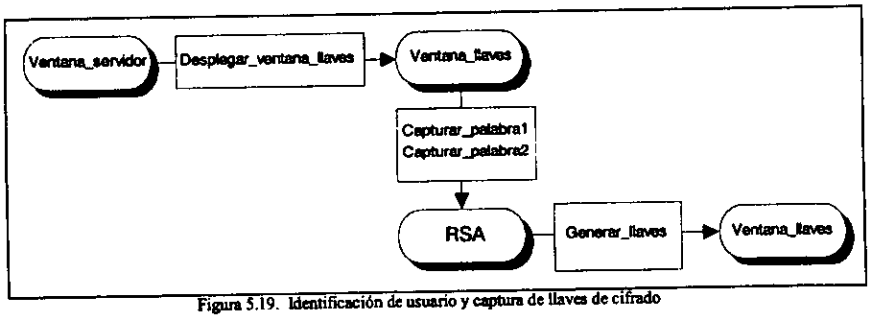


Figura 5.19. Identificación de usuario y captura de llaves de cifrado

Proceso Cifrado

El último proceso que falta por explicar es el de cifrado. Este proceso se divide en dos partes, la primera se refiere a la relación de los objetos que manejan las llaves de cifrado, y la segunda implica la descripción de las funciones que emplean los objetos. Obsérvese la figura 5.20.

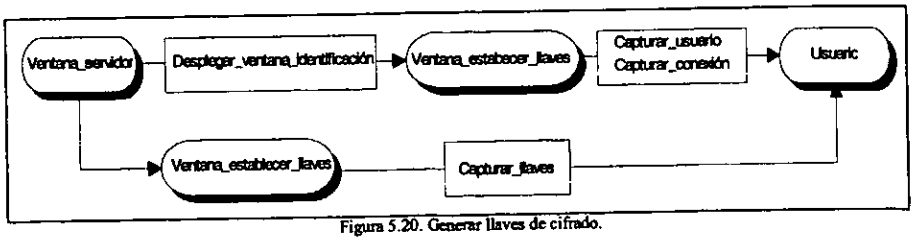


Figura 5.20. Generar llaves de cifrado.

El proceso de generar las llaves de cifrado consiste en crear la *Ventana_llaves* a partir de la *Ventana_servidor* mediante la función *Desplegar_ventana_llaves*. Una vez creada se encargará de capturar las palabras claves tecleadas por el usuario y transmitirlas al objeto *RSA* (ver códigos D.23, D.24 y D.25), el cual generará las llaves pública y privada para entregárselas a la *Ventana_llave* y desplegarlas al usuario final. Vea la figura 5.17.

A continuación se explicará en que consisten las funciones del objeto *RSA*. Realmente este objeto es una librería cuyas funciones se encargan de generar las llaves, cifrar y descifrar los datos, y la explicación de ellas se hará empleando diagramas de flujo debido al modo secuencial en que están estructuradas.

Cuando se generan las llaves de cifrado desde la función *Desplegar_llaves* del objeto *Ventana_llaves* llama a la función *Genera_Llaves* del objeto *RSA*, la cual va realizar el siguiente proceso. Se tiene como entrada una semilla que es la palabra clave que el usuario otorga, y el tiempo en que esta se hace. Con estos datos se generan tres números primos p, q y e; que a su vez sirven para generar las llaves de cifrado, llave privada (d,N) y llave pública (e,N).Obsérvese la figura 5.21.

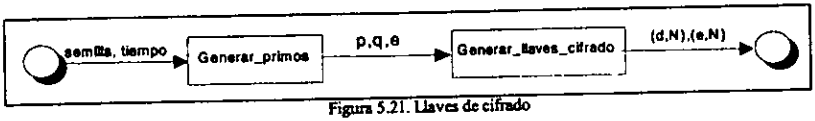


Figura 5.21. Llaves de cifrado

Ya sea en el proceso servidor o cliente, el cifrado de datos se realiza antes de ser transmitidos (ver figura 5.22). Esto ocurre en la función miembro *Enviar_mensaje* del objeto servidor y del objeto cliente (códigos D.4 y D.8).

```
Clase::Enviar mensaje {
        ...
        mensaje_cifrado=RSA->Cifra (mensaje, llave pública);
        enviar_mensaje_cifrado();
        ...
}
```

Figura 5.22. Cifrado del mensaje en las clases servidor y cliente

Obsérvese que en la figura 5.23, se tiene como entrada el mensaje a enviar y la llave pública de receptor (e,N), después se divide el mensaje en bloques $M_1, M_2, \ldots, M_n$ (el número de bloques depende del valor de N), para ser transformados en números no mayores de 32 bits. Después, estos números se cifran con la llave pública y el resultado obtenido (nc1, nc2, ... , ncn), se vuelve a transformar en bloques de caractéres ($M_{c1}, M_{c2}, \ldots, M_{cn}$). Todos los bloques de datos tienen la misma longitud y ésta debe ser menor que N (en este caso se calculó la longitud como la potencia de dos más grande que resultara menor que N, es decir $2^{\text{longitud}} < N$). El porque de esta medida se debe a que si el tamaño de N es de 32 dígitos, también lo debe ser para cada bloque de datos, ya que en el momento de cifrar o descifrar se está operando sobre un mismo tipo de datos.

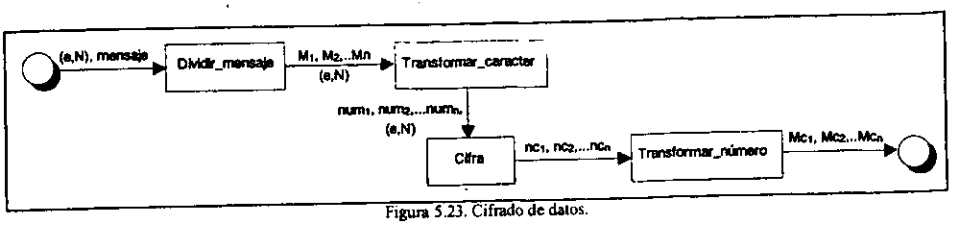


Figura 5.23. Cifrado de datos.

En el caso de que el servidor o el cliente reciban datos cifrados tendrán que descifrarlos con su llave privada (esto se realiza en la función miembro *Recibir_datos* de la respectiva clase). Observe la siguiente figura.

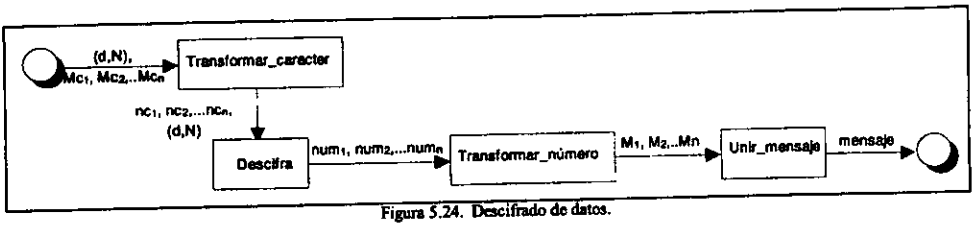


Figura 5.24. Descifrado de datos.

Una vez obtenidos los bloques de datos (M_{c1}, M_{c2}, ... ,M_{cn}) y la llave privada (d,N) se transforman los datos de tipo carácter a numérico (n_{c1}, n_{c2}, ... ,n_{cn}) y después cada uno de ellos se descifran (num_1, num_2, ... ,num_{cn}). Estos números descifrados se transforman a caracteres obteniendo los bloques M_1, M_2, ... ,M_n. Posteriormente, se insertan estos bloques para obtener el mensaje descifrado.

En el caso de que los bloques de datos no fueran del tamaño que se estipuló en un principio, se regresa una cadena nula (código D.13, *DescifraMensaje*). Esta cadena nula significa que el mensaje no se pudo descifrar, posiblemente porque se cifro con una llave pública menor o mayor a 32 bits y que no fue determinada al inicio de la conversación, la detección de este caso hace desplegar un mensaje de advertencia descrito en el siguiente seudocódigo (Figura 5.25) escrito en la función miembro *Recibir_Mensaje* de la clase servidor o cliente (códigos D.8 y D.16).

```
Clase:: Recibir_mensaje()
{   ...
 mensaje_descifrado= RSA->Descifra(Usuario->Llave_privada,Usuario->n,lpstrTexto)
 Si mensaje_descifrado == NULO entonces
          Despliega("Usuario con llave pública erronea. Posiblemente es un husmeador");
          Cierra conexión();
  ... }
```

Figura 5.25. Detección de un mensaje nulo

Un aspecto importante a considerar es la definición del tamaño de la llave. En los anteriores procesos que involucran el cifrado y descifrado se están manejando números cuyas longitudes son menores o iguales a 32 bits, esto es debido a que se está considerando que al hacer el cálculo de cifrado y descifrado se pueden obtener números hasta de 64 bits y respecto a los tipos de datos que se manejarán en el lenguaje de programación tiene como máxima longitud

de enteros hasta 64 bits. Es por ello que consideramos que la longitud de las llaves de cifrado y descifrado estén constituidas por números de 32 bits.

5.2.2.2. Interfaz de EncripData.

El diseño de la interfaz EncripData esta formada por una ventana principal llamada *Servidor*. A partir de ella se despliegan las otras ventanas de EncripData, dependiendo de la acción a realizar. El siguiente diagrama muestra el orden en que son llamadas.

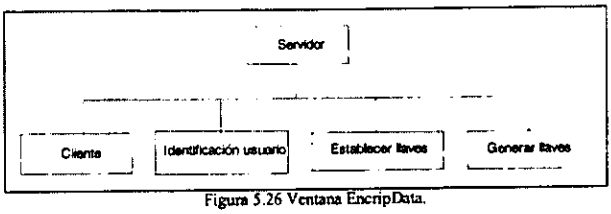


Figura 5.26 Ventana EncripData.

La interfaz de EncripData consiste en un conjunto de ventanas las cuales son: Servidor, Cliente, Identificación usuario, Establecer llaves y Generar llaves. El diseño de cada una de ellas se mostrará en las siguientes ilustraciones.

Ventana Servidor

La figura 5.27 muestra la primera ventana que se despliega cuando se ejecuta la aplicación. En ella, el usuario puede crear un grupo de conversación, o bien recibir invitaciones de otros usuarios. A partir de esta ventana se despliegan otras, así que podríamos decir que es una ventana de control.

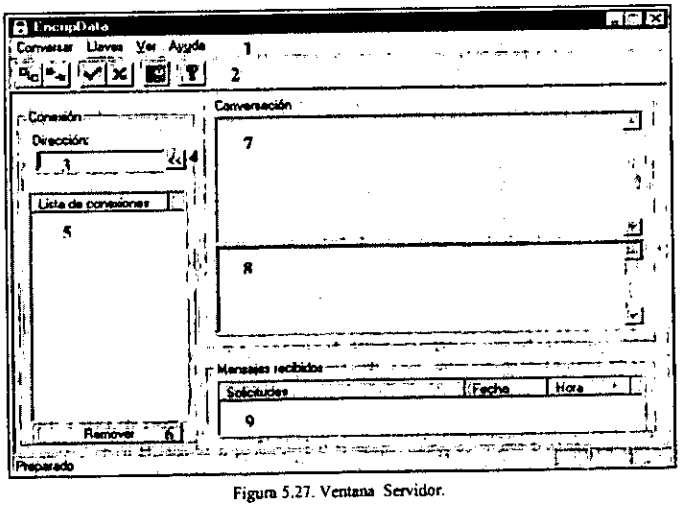


Figura 5.27. Ventana Servidor.

Descripción de los objetos de la ventana *Servidor*:

1 **Barra de menú desplegable**

- Conversación

Realizar llamada. Opción seleccionada por el usuario para iniciar un grupo de conversación. Antes de seleccionar esta opción se debe especificar los nombres de las máquinas o sus direcciones IP en la lista de conexiones en donde se encuentran los usuarios finales con los que se desea establecer una comunicación.

Desconectar usuarios. Cuando el iniciador da por finalizado el grupo de conversación utiliza esta opción para desconectar la comunicación de todos los integrantes del grupo.

Aceptar llamada. Con esta opción se acepta la petición de incorporase al grupo de conversación estableciendo un canal con el emisor.

Rechazar llamada. Esta opción es utilizada por el usuario cuando no desea incorporase al grupo de conversación creado por otro.

- Llaves

Generar. Mediante esta opción se generan las llaves de cifrado tanto pública como privada, con la opción de almacenarlas en un archivo situado en un dispositivo de almacenamiento.
Establecer. Incorpora al sistema las llaves de cifrado de un determinado usuario.
Borrar . Elimina el archivo donde se almacenan las llaves de cifrado de un determinado usuario.

- Ver

Barra de herramientas. Muestra u oculta la barra de herramientas
Barra de estado. Muestra u oculta la barra de estado.

- Ayuda

Acerca de. Muestra información del programa, número de versión y derechos de la aplicación EncripData.

2. **Barra de herramientas**. Muestra los iconos que están relacionados con la barra de menú desplegable y que anteriormente fueron señalados.
3. **Dirección**. En este cuadro de texto se teclea la dirección IP o nombre de la máquina del usuario final con el que se desea establecer una comunicación.
4. **Agregar**. Al dar un click en este botón, se agrega la dirección o nombre de máquina tecleada en el cuadro de texto *Dirección* a la lista de conexiones.
5. **Lista de conexiones**. En esta lista se enumeran las conexiones con las que el usuario (abridor del grupo de conversación) desea enviarles una solicitud de incorporación al grupo.
6. **Remover**. Con este botón el iniciador borra la lista de conexiones.

7. **Conversación del grupo.** En este cuadro de texto se despliegan todos los mensajes de los participantes en el grupo de conversación.
8. **Conversación del iniciador de grupo.** Solamente el iniciador incorpora mensajes que serán transmitidos a todos los miembros del grupo, inclusive a él mismo.
9. **Lista de llamadas.** En esta lista se enumeran las solicitudes de incorporación a un grupo de conversación.

Ventana Cliente

Esta ventana aparece cuando el usuario da un click en el botón *Aceptar llamada* desde la ventana *Servidor*. La ventana Cliente presenta dos cuadros de texto, uno que está deshabilitado y en el cual se la conversación de todos los usuarios. Y el otro cuadro de texto es donde el usuario invitado teclee su conversación (Figura 5.28).
Descripción de los objetos de la ventana Cliente:

1. **Conversación del grupo.** Aquí se despliega todos los mensajes de los participantes en el grupo de conversación.
2. **Conversación del usuario llamado.** Solamente el usuario llamado incorpora mensajes que serán transmitidos a todos los miembros del grupo, inclusive a él mismo.
3. **Desconectar.** Al dar un click en este botón el usuario llamado se desconecta del grupo de conversación.

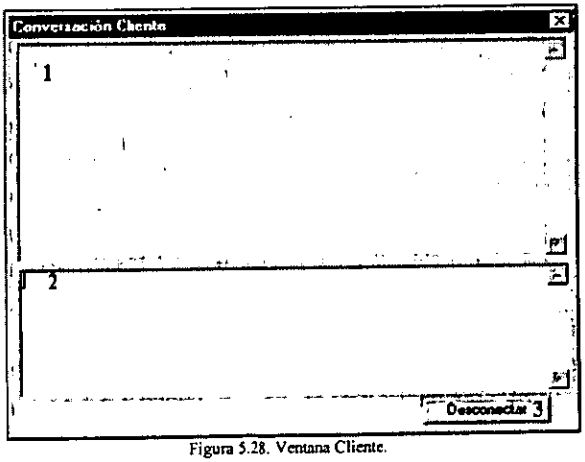


Figura 5.28. Ventana Cliente.

Ventana Generar llaves

Para crear las llaves de cifrado se selecciona del menú desplegable de la ventana *Servidor* la opción de *Llaves* y después *Generar*. A continuación se describen los objetos de la ventana Cliente:

1. **Palabra clave.** Cuadro de texto donde se teclea la palabra clave para generar las llaves pública y privada.
2. **Verificación de palabra clave.** Se vuelve a teclear la palabra clave para comprobar que es la que se había tecleado anteriormente

3. **Imprimir llaves.** Cuadro de verificación que establece si se desplegará las llaves en la sección Llave pública y privada que se encuentra en la parte inferior del cuadro de diálogo actual.
4. **Llaves en archivo.** Almacena las llaves de cifrado en un archivo junto con una palabra clave. Este archivo podrá usarlo únicamente el usuario que tenga la palabra clave.
5. **Llave pública (P) y privada (D).** Si el cuadro de verificación Imprimir llaves está seleccionado, las llaves cifrado serán mostradas en los cuadros de texto P y D.
6. **Generar.** Este botón crea las llaves de cifrado.
7. **Borrar.** Con este botón limpia todos los cuadros de diálogo y los cuadros de verificación.

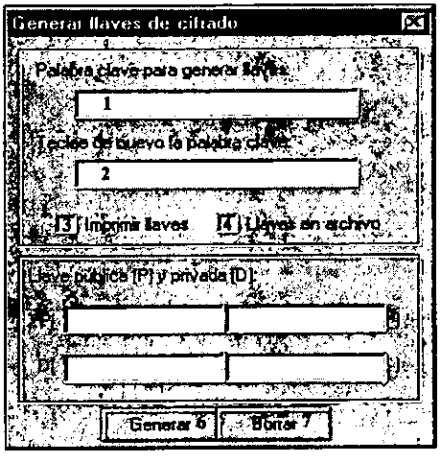


Figura 5.29. Ventana Generar llaves.

Ventana Identificación usuario

Este cuadro de diálogo aparece cuando se establece una conexión con otros usuarios por primera vez, o bien, cuando se establece que la aplicación va a ser manejada por un nuevo usuario (opción establecer del menú desplegable).

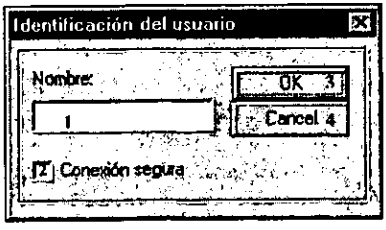


Figura 5.30. Ventana Identificación usuario

Descripción de los objetos de la ventana *Identificación usuario*:

1. **ID Usuario**. En este cuadro de texto se debe especificar el nombre que identificará al usuario durante la conversación.
2. **Conexión segura**. Este cuadro de verificación establece si los mensajes que se transmitan serán o no cifrados.
3. **Aceptar**. Botón que da por hecho que el nombre de usuario y la opción de conexión segura están correctas.
4. **Cancelar**. No almacena los datos tecleados y cierra el cuadro de diálogo.

Ventana Establecer llaves

Si en la ventana *Identificación usuario* se ha seleccionado el cuadro de verificación de conexión segura, después de dar un click al botón aceptar aparecerá otra ventana llamada *Establecer llaves* donde se deberá teclear el par de llaves de cifrado o bien seleccionar un archivo donde se encuentran almacenadas, proporcionando la palabra clave del usuario dueño de las llaves de cifrado. Si la palabra clave es la correcta, las llaves de cifrado serán agregadas a la aplicación, de otra forma se advertirá que no existirá una comunicación segura.

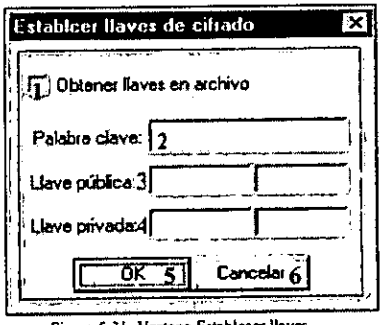


Figura 5.31. Ventana Establecer llaves.

Descripción de los objetos de la ventana *Establecer llaves*:

1. **Llaves en archivo**. Al seleccionar el este cuadro de verificación se establece que las llaves de cifrado están almacenadas en un archivo.
2. **Palabra clave**. En este cuadro de texto debe especificarse la palabra clave para obtener las llaves de cifrado en un archivo.
3. **Llave pública**. Se debe especificar la llave pública
4. **Llave privada**. Se debe teclear la llave privada
5. **Aceptar**. Mediante este botón se acepta las llaves de cifrado especificadas (mediante el teclado o en un archivo).
6. **Cancelar**. Con este botón se limpian todos los cuadros de texto y el cuadro de verificación

El diseño de los procesos que realizará EncripData ya están resueltos, faltaría escribirlos en Visual C++. El código de la aplicación se encuentra en el apéndice D, en él se podrá encontrar los procesos del servidor, cliente, identificación del usuario y cifrado.

5.2.3. Pruebas.

Considerando la figura 5.32, supongamos que el usuario A que se encuentra en W95-A desea comunicarse con el usuario B que está en la máquina W95-B, mediante EncriptData. Entonces, el usuario A incorpora en la lista de conexiones la siguiente dirección 132.248.54.60; al dar un click en el botón conectar, sí es la primera vez que ejecuta la aplicación EncriptaData, aparecerá una ventana en la cual deberá proporcionar su nombre, supongamos que sea UsuarioA, además no seleccionó el cuadro de verificación de conexión segura por lo que los datos no serán encriptados. Después de que fueron introducidos los datos en la ventana Identificación, la misma aplicación envía al usuario B una invitación que dice: "Incorporate al grupo de conversación". El usuario B acepta la invitación oprimiendo el botón aceptar (si la aplicación no tienen almacenado el nombre de usuario se desplegará la ventana Identificación) y la aplicación EncripData envía al usuario A (el iniciador del grupo de conversación) el mensaje de "Estoy en el grupo de conversación". Una vez establecida la comunicación se inicia la transmisión de mensajes entre los dos participantes del grupo de conversación.

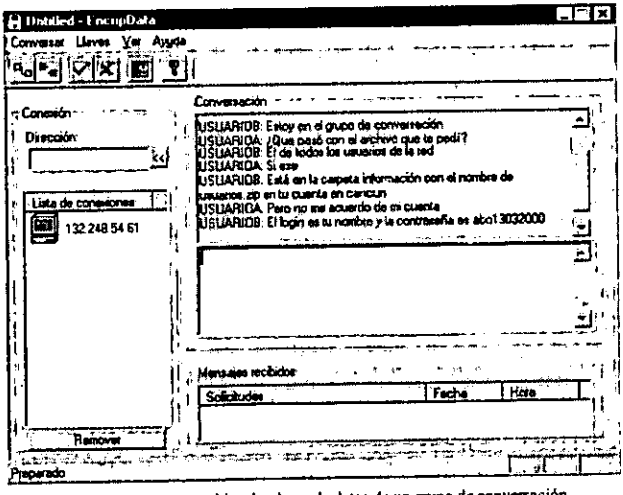


Figura 5.32. Ventana servidor desplegando datos de un grupo de conversación

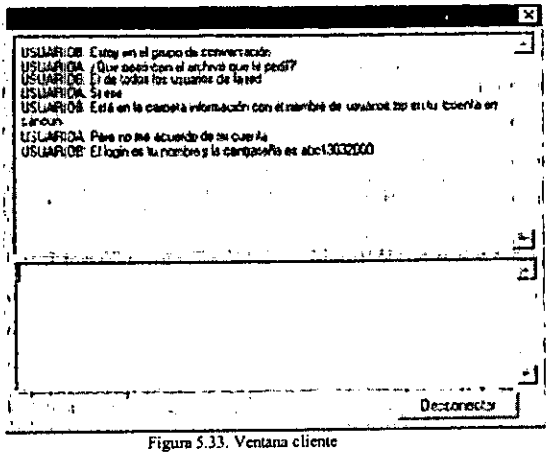


Figura 5.33. Ventana cliente

Durante la conversación el usuario A le va enviar una contraseña al usuario B. Esta contraseña es utilizada para poder accesar a un recurso. Tal vez para otros usuarios les sería muy importante obtener esa contraseña ¿sería fácil capturarla durante la conversación?

Mediante un software cuya tarea es la de analizar los paquetes que se trasmiten en la red y que puede estar al alcance de cualquier persona que navegue en Internet, se capturaron algunos paquetes durante la conversación establecida entre el usuario A y usuario B. Obsérvese figura 5.34.

```
No:                 3
Timestamp:          19:2:57:380
MAC source address: 00-00-E8-68-49-82
MAC dest address:   00-00-E8-68-32-AC
Frame:              IP
Protocol:           TCP
Source IP address:  132.248.54.60
Dest IP address:    132.248.54.61
Source port:        1025
Destination port:   700
SEQ:                23926295
ACK:                1859094
Packet size:        131

Packet data:
0000:  00 00 E8 68 32 AC 00 00 E8 68 49 82 08 00 45 F8 ...h2....hI...E.
0010:  00 75 23 1B 40 00 80 06 60 06 84 F8 36 3C 84 F8 .u#.@...'...6<..
0020:  36 3D 04 01 02 BC 01 6D 16 17 00 1C 5E 16 50 18 6=.....m....^.P.
0030:  22 38 3D 41 00 00 00 00 01 00 3E 55 53 55 41 52 "8=A......>USUAR
0040:  49 4F 41 20 3A 20 49 6E 63 6F 72 70 6F 72 61 74 IOA : Incorporat
0050:  65 20 61 6C 20 67 72 75 70 6F 20 64 65 20 63 6F e al grupo de co
0060:  6E 76 65 72 73 61 63 69 F3 6E 00 00 00 00 00 00 nversaci.n......
0070:  00 00 00 00                                      ....
```

Información del paquete

Datos del paquete

Figura.5.34. Paquete número 3 transmitido por EncripData del usuarioA.

En la figura 5.34 se muestra el contenido del paquete que se capturó. Básicamente, podemos identificar dos partes del paquete: la información y los datos. En la información del paquete encontramos, las direcciones MAC fuente y destino, direcciones IP fuente y destino, el tipo de paquete, el protocolo que se usó para empaquetarlo, el puerto fuente y destino asignados a la aplicación, los datos del paquete y otra información que se refiera al tamaño, el acuse de recibido y la secuencia del paquete. Con respecto a la segunda parte del paquete, en los datos se encuentra la información que una entidad transmite a otra.

Podemos pensar que el único interesado en la obtención del paquete es la entidad destino. ¿Y quién más estará involucrado en el proceso? La respuesta es los ruteadores, ya que se requiere que éstos obtengan la dirección destino para guiarlo. Sin embargo, también existe otra entidad que no forma parte de la comunicación directamente, me refiero al husmeador. Mediante el contenido de los paquetes el husmeador podrá enterarse de las transacciones de datos y por medio del análisis del contexto interpretar los mimos, de tal forma que pueda obtener información que le beneficie.

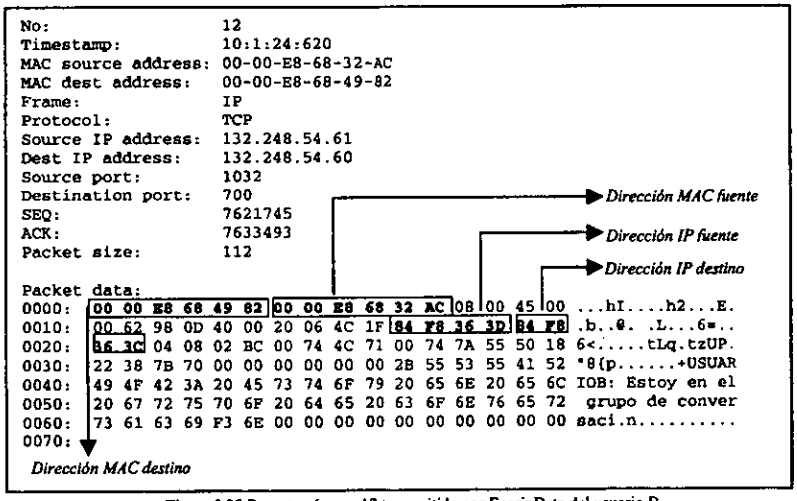
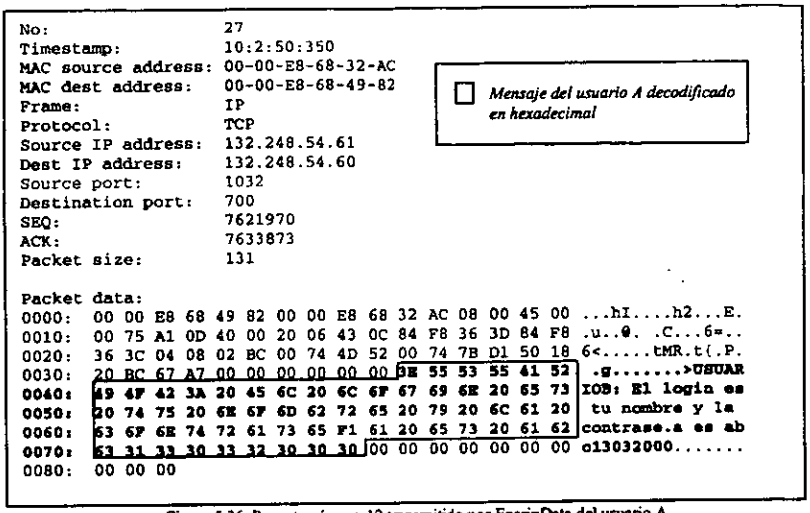


Figura.5.35 Paquete número 12 transmitido por EncripData del usuario B.

Podemos observar que en la figura 5.35, la información del paquete también se encuentra localizada en los datos codificada en hexadecimal. La dirección IP fuente está codificada como 84 F8 36 3D y la destino como 84 F8 36 3C.




```
No:                 27
Timestamp:          10:2:50:350
MAC source address: 00-00-E8-68-32-AC
MAC dest address:   00-00-E8-68-49-82
Frame:              IP
Protocol:           TCP
Source IP address:  132.248.54.61
Dest IP address:    132.248.54.60
Source port:        1032
Destination port:   700
SEQ:                7621970
ACK:                7633873
Packet size:        131

Packet data:
0000:  00 00 E8 68 49 82 00 00 E8 68 32 AC 08 00 45 00  ...hI....h2...E.
0010:  00 75 A1 0D 40 00 20 06 43 0C 84 F8 36 3D 84 F8  .u..@. .C...6=..
0020:  36 3C 04 08 02 BC 00 74 4D 52 00 74 7B D1 50 18  6<.....tMR.t{.P.
0030:  20 BC 67 A7 00 00 00 00 00 00 3E 55 53 55 41 52   .g.......>USUAR
0040:  49 4F 42 3A 20 45 6C 20 6C 6F 67 69 6E 20 65 73  IOB: El login es
0050:  20 74 75 20 6E 6F 6D 62 72 65 20 79 20 6C 61 20   tu nombre y la
0060:  63 6F 6E 74 72 61 73 65 F1 61 20 65 73 20 61 62  contrase.a es ab
0070:  63 31 33 30 33 32 30 30 30 00 00 00 00 00 00 00  c13032000.......
0080:  00 00 00
```

Figura.5.36. Paquete número 12 transmitido por EncripData del usuario A.

En el paquete número 27 mostrado en la figura 5.36, se observa que el mensaje transmitido por el usuario A está codificado en hexadecimal entre las direcciones 0030h y 0060h, también se encuentra en formato ASCII. Del contenido de este paquete podemos ver que el usuario A envía una contraseña, sí esto es detectado por un huameador, una de las opciones que tiene es extraer la contraseña, investigar el lugar, la identidad y los recursos a que tiene acceso el usuario receptor; de esta forma podrá entrar a una red ajena a él.

En cambio, si establece una conexión segura con EncripData lo anterior sería más difícil que sucediera. Veamos porque. Supongamos que tanto el usuario A, como el usuario B generan sus llaves en EncripData. El par de llaves que utiliza cada usuario son:

Usuario A [(570676801, 1318627451), (644265793, 1318627451)]

Usuario B [(650050837, 584310071), (10864813, 584310071)]

Los mensajes de ambos usuarios se cifrarán y descifrarán para ser desplegados en los respectivos destinos, así que este proceso es transparente al usuario. Pero si capturáramos los paquetes durante su transmisión, ¿cómo sería la información que llevan?

El paquete de la figura 5.37 es el primero que envía el usuario A y en él lleva el mensaje de invitación junto con la llave cifrado. Realmente la llave de cifrado no es fácil de detectar a simple vista, aunque el mensaje esta en texto claro esto no indica directamente que es la contraseña de inicio ni que método criptográfico se está utilizando.

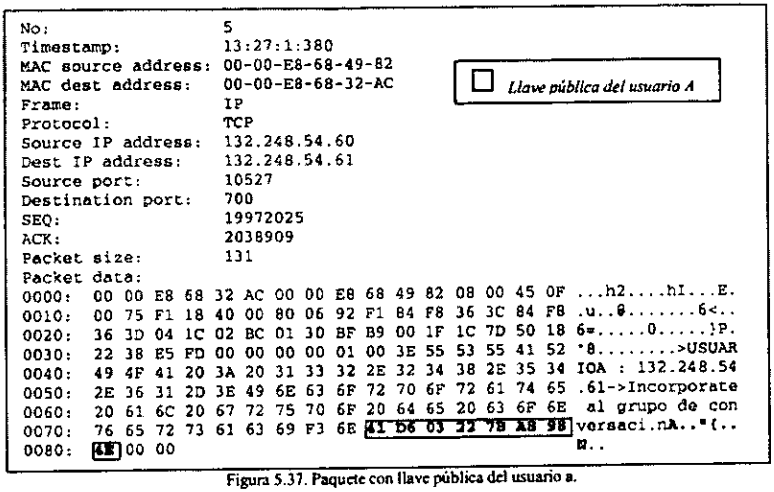


Figura 5.37. Paquete con llave pública del usuario a.

Ya establecida la comunicación se interceptaron paquetes del usuario B. En uno de ellos se almacenaba la respuesta afirmativa a la invitación del usuario A. Esta respuesta fue cifrada con la llave pública del usuario A y junto con ella fue enviada la llave pública del usuario B (Figura 5.38).

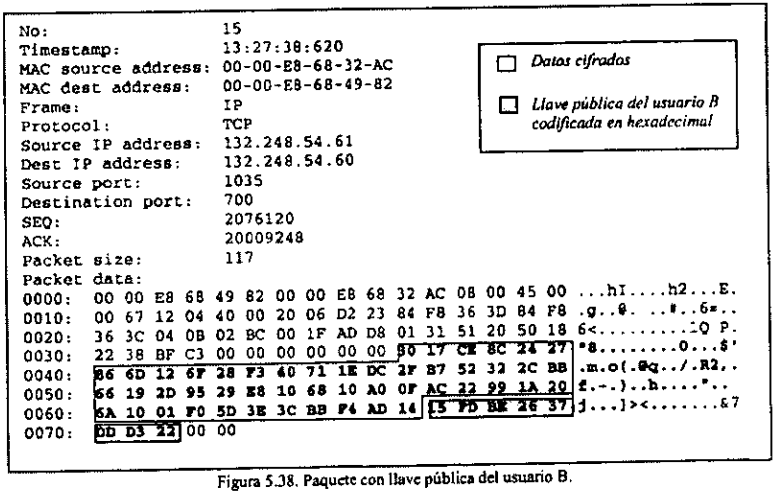


Figura 5.38. Paquete con llave pública del usuario B.

En el paquete de la figura 5.38 los datos decodificados en ASCII no forman un texto en claro. Quizá este paquete sea de interés al husmeador, pero primero tendrá que descubrir que realmente está cifrado. Para eso, tendrá que capturar más paquetes pertenecientes a la misma sesión. Después de compararlos y analizarlos, tal vez se percate de las llaves, con esta información requerirá inferir el tipo de cifrado que se utiliza para tratar de descifrar la comunicación.

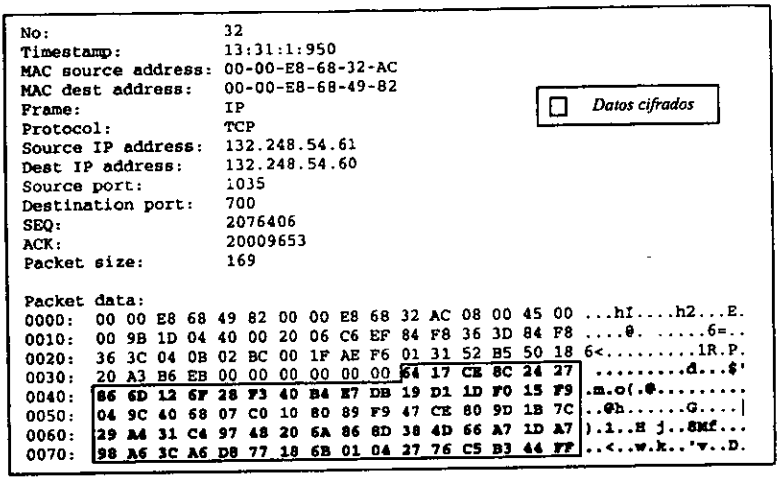


Figura. 5.39. Paquete con datos cifrados.

La figura 5.39 muestra un paquete con datos cifrados proveniente del usuario B. Cualquier persona que pueda capturar los paquetes que son transmitidos por la red sólo verá un conjunto de caracteres sin significado.

De lo anterior, podemos afirmar que la aplicación EncripData proporciona el mecanismo de confidencialidad: aunque los paquetes sean capturados por herramientas de monitoreo, éstos no son comprensibles a simple vista. Por tanto, partiendo del criterio de seguridad propuesto en el capítulo uno, la seguridad que ofrece esta herramienta es considerada de nivel tres debido a que ofrece el servicio de cifrado sobre los datos que se transmiten en las conversaciones establecidas. Además, incluye el servicio de autentificación simple ya que el emisor envía su nombre distintivo y la llave pública los cuales lo comprueban.

Otra característica importante de la aplicación es la exclusión de personas ajenas al grupo de conversación. Si un husmeador enviara un mensaje sin cifrar, éste llegaría primero al programa servidor quien se encarga de descifrarlo con la llave privada que se estableció al principio de la comunicación, pero como el mensaje no está cifrado entonces el programa envía un mensaje de alerta al usuario y no replica el mensaje a los demás participantes de la conversación. Ante esta situación podrán cerrar sus aplicaciones inmediatamente.

5.2.4. Mantenimiento.

El sistema está diseñado modularmente de tal forma que pueda ser fácilmente modificable con el fin de incorporar otros elementos que no fueron identificados en la fase de análisis de esta versión..

Para nuestro caso, una vez finalizado el sistema EncripData se observó que en el diseño (en la parte de comunicación y proceso de llaves), el cifrado y descifrado de datos se realiza mediante el empleo de llaves de 32 bits, recordando que la vulnerabilidad de este algoritmo de cifrado es el tamaño de las llaves, por lo que es posible escalar el tamaño de la llave dependiendo del nivel de seguridad que se desee alcanzar. Para esto, en el código de programación sólo se tendrá que modificar la librería RSA sin alterar otras partes del sistema.

También, se podría crear otra librería con otro tipo de algoritmo de encriptamiento; sin embargo se tendrían que rediseñar y programar las partes que únicamente utilizan las funciones de RSA en la aplicación EncripData. En el crecimiento de esta herramienta también se podría incluir la transferencia de archivos, donde el volumen de información a manejar será mayor.

Conclusiones

De lo presentado en este trabajo podemos concluir que la seguridad a nivel mundial es un fenómeno que tomó por sorpresa a toda la comunidad de telecomunicaciones ya que es un ámbito de trabajo abierto y desprotegido. En el área de las redes de computadoras, la criptografía está siendo la herramienta más utilizada para implementar mecanismos que protejan la transmisión de datos y los recursos computacionales de entidades que así lo requieran. A partir del estudio de ellas, se puede observar que en este campo se requieren de recursos humanos con una fuerte formación en las ciencias de la computación y las matemáticas.
La aplicación desarrollada para resolver el problema de comunicación presentado en el Centro de Instrumentos de la UNAM, es una herramienta de seguridad que contiene las siguientes características:

- Mediante EncripData se establece una conexión punto a punto que permite la comunicación en línea.
- La confidencialidad de la información se proporciona a través del mecanismo de cifrado de llave pública RSA.
- La autentificación que se maneja en la aplicación es de tipo débil.
- Es clasificada como de nivel tres debido a que maneja servicios de cifrado y autentificación.

De lo anterior, se proponen las siguientes modificaciones para futuras versiones de EncripData. Incrementar la longitud de las llaves de cifrado con el fin de hacerla más resistente a un ataque de criptoanálisis; implementar servicios de autentificación fuerte e incluir la transferencia de archivos.
Finalmente se generó material bibliográfico que puede servir como un auxiliar en la materia de redes de computadoras que se imparte en esta Facultad.

Bibliografía

1. Tanenbaum, ***Redes de Computadoras***, Prentice Hall, 3ª ed, México 1997.
2. William Stallings, ***Crytography and Network Security. Principles ans Practice***, Prentice Hall, 2ª ed, United States of America 1999.
3. Schneier, ***Applied Criptography. Protocols, Algorithms, and Source Code in C***, Wiley, 2ª ed, United States of America 1996.
4. Farley, ***LAN TIMES. Guide to Security and Data Integrity***, McGraw-Hill ,United States of America 1996.
5. Kruglinski, ***Programación Avanzada con Visual C++***, España 1998.
6. García, Ferrando, ***Redes de Alta Velocidad***, España 1997.
7. Heywood, Scrimger, ***Networking with Microsoft TCP/IP, Certified Administrator's Resource Edition***, United States of America 1997.
8. Sommerville, ***Software Engineering***, McGraw-Hill, 5ª ed. United States of America 1997.
9. Faireley, ***Ingeniería de Software***, McGraw-Hill, United States of America 1987.
10. RFC 1001, ***Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods***, Marzo 1987.
11. RFC 1002, ***Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications***, Marzo 1987.