CAPÍTULO 1

ANTECEDENTES

ANTECEDENTES

De acuerdo con el diccionario de la lengua española, un sistema se define como un "conjunto de cosas que relacionadas entre sí ordenadamente contribuyen a determinado objeto". Un sistema de defensa para una red, por lo tanto, es una serie de elementos relacionados entre sí de manera ordenada que tiene como objetivo proteger los activos de una red, y en este trabajo de tesis, los elementos son las herramientas de *software libre*.

En este capítulo se hace una recopilación de los conocimientos previos sobre las redes de computadoras que permiten comprender en lo subsecuente tanto términos como conceptos utilizados a lo largo de este trabajo.

1.1 DESARROLLO DEL *SOFTWARE LIBRE*

El *software libre* ha contribuido ampliamente en el desarrollo de diversos sistemas operativos y aplicaciones, así como en el desarrollo de Internet. Actualmente se crean sistemas muy avanzados y elaborados, bajo este concepto, sistemas operativos, librerías, juegos, bases de datos, procesadores de texto, y diversos programas especializados han sido desarrollados por un número cada vez más grande de comunidades de programadores de *software libre*.

Tanto los usuarios finales como los programadores contribuyen a mejorar el rendimiento de este tipo de programas, crean discusiones en foros, avisan sobre ciertos fallos, y mejoran el código fuente permanentemente, lo que ha conducido a la rápida expansión del movimiento de libre distribución. Algunos programas pueden ser adquiridos a cambio de una retribución económica, otros se adquieren de manera totalmente gratuita y el factor más importante es que se adquiere el código fuente y la libertad de usarlo y adaptarlo a nuestras propias necesidades. Esto tiene que ver con la manera en que surgió el movimiento de *software libre*, y con el concepto de libertad que se maneja en este movimiento.

1.1.1 Historia de la libre distribución

El movimiento del *software libre* o software de libre distribución tiene su origen en el nacimiento de Unix, razón por la que se asocia a la libre distribución con los sistemas Unix y Linux, aún cuando el concepto se ha extendido para casi todos los sistemas operativos de computadoras disponibles.

El sistema operativo Unix fue inventado por los laboratorios Bell, una división de investigación de AT&T a fines de los años sesenta. Poco tiempo después, AT&T permitió a las universidades utilizar su software. Como AT&T estaba regulado, no podía hacer negocios vendiendo Unix, así que ofreció a las universidades el código fuente para sus sistemas operativos. Éstas comenzaron a idear sus propias adiciones y modificaciones para

el código AT&T original. Algunas sólo realizaron cambios mínimos. Otras, como la universidad de Berkeley en California, realizaron tantas modificaciones que crearon un tipo totalmente nuevo del código. Pronto, el campo Unix se dividió en dos: el código AT&T, o mini, y el código base BSD, que generó muchas de las versiones Unix de libre distribución basadas en BSD actuales.

En un principio, los programadores que se desempeñaban en ámbitos empresariales y universitarios, creaban y compartían software sin ningún tipo de restricciones; esto cambió en la década de los 80, cuando las computadoras más modernas comenzaron a utilizar sistemas operativos privativos, forzando a los usuarios a aceptar condiciones restrictivas que impedían realizar modificaciones a dicho software, incluso los programadores tenían que firmar acuerdos de no revelación, con lo cual se impedía la colaboración entre desarrolladores de software.

En 1983, Richard Stallman inició el proyecto GNU (GNU is Not Unix), proyecto de desarrollo de *software libre*, como una forma de eliminar los obstáculos impuestos por los dueños del software privativo y de devolver el espíritu cooperativo que prevalecía en la comunidad computacional en sus inicios. De inmediato se pensó que esta nueva comunidad necesitaría una base sobre la cual pudiera trabajar, un sistema operativo de libre distribución, así fue como Stallman y los desarrolladores del proyecto GNU decidieron crear un sistema operativo libre, cuyo código fuente fuese accesible para cualquier persona, y además tuviera la libertad de usarlo con cualquier propósito, hacerle mejoras, adaptarlo a sus necesidades, copiarlo y distribuirlo.

El nuevo sistema operativo debía ser compatible con Unix porque dicho sistema ya estaba probado y era portable, así sería muy fácil para los usuarios cambiar de sistema operativo al nuevo GNU. El editor GNU Emacs fue una de las primeras tareas del nuevo proyecto GNU, para septiembre de 1985, GNU Emacs ya era usable y se distribuía mediante un pago. De esta manera se inició el negocio de distribución de *software libre*. Poco después, dio comienzo el desarrollo de un nuevo compilador, el GCC (GNU Compiler Collection). Ésta fue una de las herramientas más importantes del proyecto, GNU desarrolló un potente compilador en lenguaje C que siguen usando los sistemas operativos Linux hasta nuestros días.

Stallman creó en 1985 la Free Software Foundation, una organización libre de impuestos para el desarrollo de *software libre*. Esta organización se dedicó a implementar las utilidades que ofrecía Unix para el nuevo sistema operativo libre. A medida que el proyecto GNU crecía, los componentes del nuevo sistema operativo seguían en desarrollo, pero la meta de desarrollar por completo el sistema GNU todavía no se alcanzaba. Cada

componente de un sistema GNU se implementó en un sistema Unix, y para 1990 se tenía el sistema casi completo, el único componente importante que faltaba era el núcleo.

En 1991, Linus Torvalds, un estudiante universitario finlandés, hizo la aportación de la pieza que faltaba para tener un sistema operativo completo. Torvalds quería ejecutar una versión de Unix en su computadora, ya que éste era el sistema que se usaba en la universidad, para ello compró MINIX, una versión de computadora simplificada del sistema operativo Unix. Torvalds se sintió frustrado por las limitaciones de MINIX, particularmente en el área de la emulación de terminal y comenzó a trabajar para crear un programa emulación de terminal de computadora. Cuando terminó su programa comprendió que había creado un núcleo de sistema operativo y lo denominó Linux, pronto lo distribuyó a algunos grupos de noticias de USENET y los usuarios empezaron a sugerir mejoras y complementos. El nuevo núcleo fue adaptado al ambiente de GNU, y esto creó un amplio espectro de aplicaciones para el nuevo sistema operativo de libre distribución, resultado de la unión del software del proyecto GNU, múltiples programas de *software libre* y el núcleo Linux.

La designación Linux fue usada al principio únicamente para el núcleo, sin embargo, dicho núcleo era usado con frecuencia junto con otro software, especialmente del proyecto GNU. Richard Stallman fundador de GNU solicitó que el nombre GNU/Linux fuera usado para reconocer el rol del software GNU, sin embargo el nombre Linux es comúnmente utilizado para designar al sistema operativo completo.

La combinación de las comunicaciones globales de bajo costo y la facilidad del acceso a la información a través de las páginas web originaron un renacimiento en la innovación y el desarrollo del mundo de la libre distribución. Ahora, los programadores podían colaborar instantáneamente y colocar sus sitios web detallando su trabajo para que cualquiera pudiese encontrarlo utilizando motores de búsqueda. Los proyectos de trabajo en rutas paralelas combinaron sus recursos y esfuerzos, surgieron otros grupos a partir de los grupos más grandes, confiando en que ahora podían apoyar sus esfuerzos.

Actualmente existen muchos desarrolladores de *software libre* para distintas plataformas y este software se utiliza incluso junto con software privativo. Sistemas operativos, aplicaciones y librerías, se distribuyen de manera gratuita o a cambio de cierta remuneración en Internet. GNU/Linux es un gran ejemplo del desarrollo del *software libre*, sin embargo no es el único, infinidad de programas se realizan bajo este concepto y cada día se disputan un lugar en el ámbito computacional con software propietario.

1.1.2 La libertad del *software libre*

En su sitio web (http://www.gnu.org/philosophy/free-sw.es.html) el proyecto GNU define una serie de elementos que permiten conocer el concepto de libertad y sus alcances, cuando se habla de *software libre*[1].

El *software libre* es una cuestión de la libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Más precisamente, significa que los usuarios de programas tienen las cuatro libertades esenciales.

- La libertad de ejecutar el programa, para cualquier propósito (libertad 0).
- La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (la 3ª libertad). Si lo hace, puede dar a toda la comunidad una oportunidad de beneficiarse de sus cambios. El acceso al código fuente es una condición necesaria para ello.

Que un programa sea *software libre* no significa que no sea comercial. Un programa libre debe estar disponible para el uso comercial, la programación comercial y la distribución comercial.

El proyecto GNU surgió con la idea principal de dar libertad a los usuarios de *software libre* y que dicho software no se convirtiera en software privativo, dichas libertades se sintetizaron en una licencia especial, la licencia GPL (General Public License). Esta licencia es un conjunto específico de términos de distribución empleados para proteger un programa con copyleft. El Proyecto GNU utiliza esta licencia para la distribución de la mayoría del software de GNU.

El copyleft usa la ley de copyright, pero sirviendo para un propósito opuesto al usual, en lugar de ser un medio de privatizar el software, se transforma en un medio de mantener libre al software. La idea central es dar el permiso para correr el programa, modificarlo, copiarlo y redistribuir versiones modificadas, pero sin permiso para agregar restricciones propias. De esta manera las libertades básicas que definen al *software libre* quedan garantizadas para que cualquier persona tenga una copia y a su vez, estas libertades se convierten en derechos inalienables.

1.2 SISTEMAS OPERATIVOS DE LIBRE DISTRIBUCIÓN

El desarrollo del software ha dado lugar a que surjan nuevos sistemas operativos libres, con características diversas y licencias diferentes, unas más restrictivas que otras.

Existe una gran cantidad de sistemas operativos libres para diferentes arquitecturas, fundamentalmente se encuentran los sistemas operativos basados en BSD, en Linux y otras distribuciones libres, a continuación se describen las características más importantes de algunos de ellos.

1.2.1 Distribuciones basadas en BSD

OpenBSD es un sistema operativo libre tipo Unix, multiplataforma, descendiente de NetBSD, con especial atención en la seguridad y la criptografía. Su filosofía se resume en tres palabras, "libre, funcional y seguro". Se considera uno de los sistemas operativos más seguros y estables, aunque en su instalación por defecto se activen la menor cantidad de servicios posibles, esto también se considera como una práctica de seguridad. Utiliza un algoritmo de cifrado de contraseñas que hace muy difícil el procesamiento en paralelo y por lo tanto, también los intentos de descifrado. Debido a las características de este sistema operativo, se utiliza mucho en el sector de la seguridad informática como sistema operativo base para implementar *firewalls* y sistemas de detección de intrusos.

FreeBSD es un sistema operativo multitarea, multiusuario y multiproceso para procesadores de arquitectura Intel y compatibles con Intel, como AMD y Cyrix. También es posible utilizarlo hasta en otras arquitecturas como Alpha, AMD64, MIPS, PowerPC y UltraSPARC. Está hecho para ser compatible con la norma *Posix*, al igual que varios otros sistemas clones de Unix. El sistema incluye el núcleo, la estructura de archivos del sistema, bibliotecas de la API de C, y algunas utilerías básicas. FreeBSD es compatible con binarios de varios sistemas operativos del tipo Unix, incluyendo Linux, la razón de esto es la necesidad de ejecutar aplicaciones no libres desarrolladas para Linux, que por las ventajas que ofrecen, resulta conveniente portarlas a FreeBSD.

NetBSD es un sistema operativo tipo Unix, libre, disponible para más de 50 plataformas hardware, desarrollado a partir una gran variedad de software que incluye a 4.4BSD Lite de la Universidad de California-Berkeley, Net/2 (Berkeley Networking Release 2), el sistema X de ventanas y software GNU. Su principal ventaja es ofrecer un sistema operativo estable, multiplataforma, seguro y orientado a la investigación. Está diseñado teniendo como prioridad escribir código de calidad y bien organizado, y teniendo muy en cuenta el cumplimiento de estándares.

1.2.2 Distribuciones de GNU/LINUX

Debian GNU/Linux es un sistema operativo libre que se caracteriza por su portabilidad, su versión estable tiene soporte para 11 plataformas, una variedad muy amplia de software disponible y un grupo de herramientas que facilitan el proceso de instalación y actualización. Los sistemas Debian usan el núcleo de Linux. La mayoría de las herramientas básicas que completan el sistema operativo, provienen del proyecto GNU, por supuesto, herramientas de libre distribución. Además viene con más de 20,000 paquetes, es decir, software precompilado para una instalación sencilla en la máquina, todo de libre distribución.

Fedora es una distribución de Linux, libre, de propósitos generales y de código abierto. Destaca la implementación de una gran variedad de políticas de seguridad, como SELinux (Security Enhanced Linux), una colección de programas que modifican el núcleo fortaleciendo los mecanismos de control de acceso y forzando la ejecución de procesos dentro de un entorno con los mínimos privilegios necesarios. Incluye el control de acceso obligatorio (Mandatory Access Control), un elemento de seguridad que, a través de los módulos de seguridad de Linux que están en el kernel del sistema, evita el uso no autorizado de un recurso y el uso de un recurso de manera no autorizada.

Fedora soporta las arquitecturas x86, x86-64 y powerPC. Está diseñado de tal manera que se facilitan las tareas de instalación y configuración ya que incluye instaladores y herramientas gráficas. Sólo contiene una pequeña selección de paquetes de software, pero existen muchos almacenes disponibles para completar esta distribución.

Ubuntu es una distribución Linux que ofrece un sistema operativo enfocado a computadoras de escritorio aunque también proporciona soporte para servidores. Es una de las más importantes distribuciones de GNU/Linux a nivel mundial. Ubuntu es una distribución basada en Debian GNU/Linux y soporta las arquitecturas x86 y AMD64, aunque ha sido portada a otras cinco arquitecturas. Esta distribución ha sido traducida a numerosos idiomas y sus desarrolladores se basan en el trabajo de las comunidades de Debian, GNOME y KDE.

OpenSUSE es una distribución basada en Linux, auspiciada por Novell y AMD, completamente de código abierto, los desarrolladores de OpenSUSE han centrado sus esfuerzos en migrar ReiserFS a ext3 como sistema de archivos y en incluir soporte legal de MP3 y mejoras en los tiempos de carga. Novell continúa el desarrollo a puerta cerrada de dos distribuciones dedicadas al ámbito empresarial, SUSE Linux Enterpries Desktop y SUSE Linux Enterprise Server.

1.3 LAS REDES DE COMPUTADORAS

Una red de computadoras es un sistema de interconexión entre computadoras que permite compartir recursos e información[2]. En una red de computadoras se puede compartir información y recursos tanto de hardware como de software, lo que supone muchas ventajas para sus usuarios dependiendo de la finalidad con la que se haya diseñado la red.

A continuación se abordan diferentes clasificaciones de redes, de acuerdo con diferentes parámetros; también se tratan temas que tienen que ver con la manera en que se reconoce un equipo en una red, cómo se asignan las direcciones y que máscaras de subred se utilizan en las redes.

1.3.1 Diferentes clasificaciones de las redes de computadoras

Dependiendo cobertura geográfica:

- Si se conectan todas las computadoras dentro de un mismo edificio se denomina LAN (Local Area Network).
- Si se encuentran en edificios diferentes distribuidos en distancias no superiores al ámbito urbano, se denomina MAN (Metropolitan Area Network).
- Si están instaladas en edificios diferentes de la misma o distinta localidad, provincia o país, se denomina WAN (Wide Area Network).

Las redes de computadoras también se pueden clasificar de acuerdo con su topología. Se denomina topología a la forma geométrica en que están distribuidas las estaciones de trabajo y los cables que las conectan.

Las estaciones de trabajo de una red se comunican entre sí mediante una conexión física y el objeto de la topología es buscar la forma más económica y eficaz de conectarlas para, al mismo tiempo, facilitar la fiabilidad del sistema, evitar los tiempos de espera en la transmisión de los datos, mejorar el control de la red y permitir de forma eficiente el aumento de las estaciones de trabajo.

Las formas más utilizadas son:

Configuración en bus

En ella todas las estaciones comparten el mismo canal de comunicaciones, toda la información circula por ese canal y cada una de ellas recoge la información que le corresponde, véase la figura 1.1, Configuración en bus. Esta configuración es fácil de instalar, la cantidad de cable a utilizar es mínima, tiene una gran flexibilidad a la hora de

aumentar o disminuir el número de estaciones y el fallo de una estación no repercute en la red, aunque la ruptura de un cable la dejará totalmente inutilizada.

Entre sus inconvenientes destacan la facilidad de intervenir este tipo de redes, por usuarios de fuera de la red; su longitud no puede sobrepasar los 2000 metros. Además, el control del flujo se dificulta, ya que aunque varias estaciones intenten transmitir a la vez, como hay un único bus, sólo una de ellas podrá hacerlo, por lo que entre más estaciones tiene la red, es más complicado el control del flujo. Es la configuración más extendida actualmente y es usada por la red Ethernet.

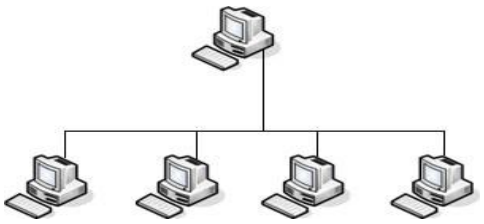

Figura 1.1, Configuración en bus

Configuración de anillo

En ella todas las estaciones están conectadas entre sí formando un anillo, de forma que cada estación sólo tiene contacto directo con otras dos, como se muestra en la figura 1.2, Configuración de anillo. En las primeras redes de este tipo los datos se movían en una única dirección, de manera que toda la información tenía que pasar por todas las estaciones hasta llegar a la estación destino. Las redes más modernas disponen de dos canales y transmiten en direcciones diferentes por cada uno de ellos. Este tipo de redes permite aumentar o disminuir el número de estaciones sin dificultad; pero, a medida que aumenta el flujo de información, será menor la velocidad de respuesta de la red.

Esta configuración tiene algunos inconvenientes, entre ellos, el que un fallo en una estación puede dejar bloqueada la red. También, un fallo en un canal de comunicaciones la dejará bloqueada en su totalidad y será bastante difícil localizar el fallo y repararlo de forma inmediata. Aunque su instalación es compleja, su uso está extendido por el entorno industrial.

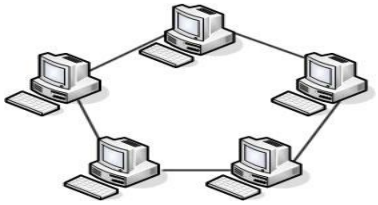

Figura 1.2, Configuración de anillo

Configuración en estrella

Esta forma de configuración es una de las más antiguas. Todas las estaciones están conectadas directamente al servidor y todas las comunicaciones se han de hacer necesariamente a través de él, como se muestra en la figura 1.3, Configuración en estrella. Permite incrementar y disminuir fácilmente el número de estaciones, si se produce un fallo en una de ellas no repercutirá en el funcionamiento general de la red; pero si se produce un fallo en el servidor, la red completa se vendrá abajo. Tiene un tiempo de respuesta rápido en las comunicaciones de las estaciones con el servidor, pero es más lento cuando se establece comunicación entre las distintas estaciones de trabajo.

La configuración en estrella no es muy conveniente para grandes instalaciones y su costo es caro debido a la gran cantidad de cableado y a la complejidad de la tecnología que se necesita para el servidor.

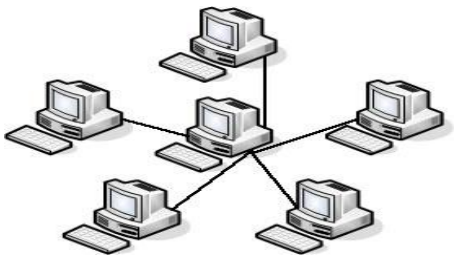

Figura 1.3, Configuración en estrella

Configuración híbrida

En esta configuración son diversas las formas en que se pueden combinar las topología mencionadas, entre las cuales se encuentra la configuración mixta en estrella/bus, en la cual un multiplexor de señal ocupa el lugar de la computadora central de la configuración en estrella, estando determinadas estaciones de trabajo conectadas a él, y otras conectas en bus junto con otro u otros multiplexores, una forma de configuración híbrida es la que se muestra a continuación en la figura 1.4, Configuración híbrida.

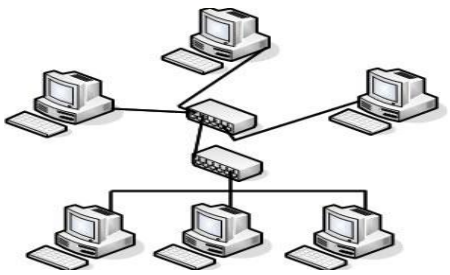

Figura 1.4, Configuración híbrida

1.3.2 Denominación de un equipo en TCP/IP

Debido a las múltiples ventajas que ofrece Internet, cada vez más redes de computadoras tienen una conexión a la red de redes. Esto ha hecho que Internet crezca de manera inusitada, y que existan millones de equipos activos conectados de manera temporal o permanente, equipos que tienen que ser identificados de alguna manera.

Es importante que se establezca la identificación del equipo que forma parte de una red, de una forma que evite su duplicidad dentro de todas las computadoras que puedan conectarse. Para ello, en TCP/IP se utiliza el nombre del usuario y el nombre del dominio de la red. Para identificar al usuario es necesario nombrarlo evitando que pueda haber dos con el mismo nombre y produzca confusiones al servidor de la red.

Para identificar a la red se utiliza el concepto de dominio. La estructura del dominio se asemeja a un árbol invertido, es decir, el tronco se encuentra en la parte superior y las ramas en la parte inferior, y cada hoja corresponde a un dominio. La identificación de un dominio está formada por varios apartados separados por un punto. Cada uno de ellos recibe el nombre de subdominio. El subdominio situado más a la derecha es el de carácter más general y recibe el nombre de dominio de nivel alto.

El nombre de un dominio completamente calificado (FQDN, Full Qualified Domain Name) comienza con el nombre de la estación de trabajo o host, un punto, y el nombre de la red (Dominio). Por ejemplo, si se denomina a la computadora como COMP001 y a la red principal como RED1, la identificación completa de la estación de trabajo sería COMP001. RED1. Si, a su vez, esta red formara parte de otra red superior, se volvería a poner otro punto y el nombre de dicha red, por ejemplo, COMP001.RED1.MEC; después del host vendría el o los subdominios y, para finalizar, el dominio.

También es interesante identificar a la institución de la que forma parte la red, así como la organización o el país a la que pertenece, en la tabla 1.1, Dominio de alto nivel de organización, se muestran las abreviaturas para los dominios de alto nivel de organización. Estos dos nuevos conceptos se añaden separados por puntos. Por ejemplo, para una red que pertenece a una institución educativa mexicana, se tendría la terminación .edu.mx.

Tabla 1.1 Dominio de alto nivel de organización

Dominio	Significado
com	Organización comercial
edu	Institución educativa
gov	Institución gubernamental
int	Organización internacional
mil	Organización militar
net	Organización de red
org	Organización sin ánimo de lucro
mx	Organización mexicana

1.3.3 Direcciones IPV4

Una dirección IP es un número binario de 32 bits utilizado para identificar excepcionalmetne al host y a su red[3].

Las direcciones *IP* consiguen que el envío de datos entre computadoras se realice de forma eficaz, de forma parecida a como se utilizan los número de teléfono en las llamadas telefónicas. La dirección del equipo indica el número que corresponde a la computadora dentro de la red.

Actualmente, las direcciones *IP* de la versión actual (Ipv4) tienen 32 bits, formados por cuatro campos de 8 bits cada uno, separados por puntos, por tanto, las direcciones *IP* están representadas en forma binaria. Cada uno de los campos de 8 bits puede tener un valor que esté comprendido entre 0 y 255 decimal. Normalmente y debido a la dificultad del sistema binario, la dirección *IP* se representa en decimal. Los cuatro octetos de la dirección *IP* componen una dirección de red y una dirección de equipo que están en función de la clase de red correspondiente.

Existen cinco clases de redes: A, B, C, D o E, esta diferencia está dada en función del número de computadoras que va a tener la red[4].

La clase A contiene 7 bits para la dirección de red, el primer bit del octeto siempre es un cero, y los 24 bits restantes representan a direcciones de equipo. De esta manera, se puede tener un máximo de 128 redes, aunque en realidad se tienen 126, ya que están reservadas las redes cuya dirección de red empieza por cero y por 127, cada una de las cuales puede tener 16,777,214 computadoras ya que se reservan aquellas direcciones de equipo en binario, cuyos valores sean todos ceros o todos unos. Las direcciones en representación decimal están comprendidas entre 0.0.0.0 y 127.255.255.255 y la máscara de subred es 255.0.0.0.

La clase B contiene 14 bits para direcciones de red, ya que el valor de los dos primero bits del primer octeto es siempre 10 y 16 bits para direcciones de equipo, lo que permite tener un máximo de 16,384 redes, cada una de las cuales puede tener 65,536 equipos, aunque en realidad tienen 65,534 cada una, ya que se reservan aquellas direcciones de equipo, en binario, cuyos valores sean todos ceros o todos unos. Las direcciones, en representación decimal están comprendidas entre 128.0.0.0 y 191.255.255.255 y su máscara de subred es 255.255.0.0.

La clase C contiene 21 bits para direcciones de red, ya que el valor de los tres primero bits del primer octeto ha de ser siempre 110 y 8 bits para direcciones de equipo, lo que le permite tener un máximo de 2,097,152 redes, cada una de las cuales puede tener 256 equipos, aunque en realidad tienen 254 equipos cada una, ya que se reservan aquellas direcciones de equipo, en binario, cuyos valores sean todos ceros o todos unos. Las direcciones, en representación decimal, están comprendidas entre 192.0.0.0 y 223.255.255.255. y su máscara de subred es 255.255.255.0.

La clase D se reserva todas las direcciones para multidestino o multicasting, es decir, una computadora transmite un mensaje a un grupo específico de computadoras de esta clase. El valor de los cuatro primeros bits del primer octeto ha de ser siempre 1110 y los últimos 28 bits representan los grupos multidestino. Las direcciones, en representación decimal, están comprendidas entre 224.0.0.0 y 239.255.255.255.

La clase E se utiliza con fines experimentales únicamente y no está disponible para el público. El valor de los cuatro primeros bits del primer octeto ha de ser siempre 1111 y las direcciones, en representación decimal, están comprendidas entre 240.0.0.0 y 255.255.255.255.

1.4 SERVICIOS DE RED

El objetivo de una red es que los usuarios de la misma puedan compartir recursos e información, y así mejorar el rendimiento global de la organización. El uso de la red y los servicios que esta ofrece proporcionan una serie de ventajas, como la facilidad en la comunicación, una mejora en la competitividad, reducción de costos, mejoras en la administración, mejoras en integridad de datos y seguridad de la información.

Las redes con una arquitectura cliente-servidor permiten la distribución de las tareas entre programas que se ejecutan en el servidor o en la terminal del usuario, el agente que requiere que se haga determinada tarea se llama cliente y el agente que realiza dicha tarea se llama servidor. Una arquitectura de este tipo permite mejorar el rendimiento, reduce los costos, al compartir recursos, y facilita la administración, al concentrarse los trabajos en los servidores.

Existen múltiples recursos que pueden ser compartidos en una red para beneficio de sus usuarios. Algunos recursos proporcionan servicios que se identifican como servicios de hardware y otros como servicios de software, por ejemplo, el uso de un servidor de impresión se identifica como un servicio de hardware, mientras que el uso de un programa como el servidor *apache*, se identifica como un servicio de software. A continuación se describen algunos servicios de software que se ofrecen en la mayoría de las redes.

Uno de los servicios básicos que ofrecen todas las redes de computadoras es el control de acceso, que comprende tanto los elementos de verificación de la identidad de los usuarios como los servicios y herramientas necesarios para delimitar el uso de los recursos por los usuarios. Este servicio implementa mecanismos que permiten establecer permisos para la ejecución de tareas dentro de la red, acceso a directorios y archivos y el uso adecuado de los recursos de la red.

Otro servicio básico es el almacenamiento. Ofrecer una cierta capacidad de almacenamiento significa implementar los mecanismos de seguridad e integridad en la información y configurar el sistema de tal manera que el usuario pueda disponer de sus datos en cualquier momento. Este servicio permite disminuir las capacidades de almacenamiento de las terminales del usuario.

La ejecución de programas especializados y el uso de recursos de gran capacidad son algunos de los servicios más importantes para los usuarios de una red y representan una de las principales ventajas de implementar una red. Si todos los usuarios de la red pueden hacer uso de la capacidad de cálculo de los equipos de la red y ejecutar programas especializados que requieren de procesadores potentes, no necesitan hacer instalaciones de gran capacidad en sus terminales ni requieren la instalación de software especializado para

desempeñar su trabajo, de esta manera se facilita el trabajo en la organización y se reducen costos al utilizar la red como una entidad única.

Un aspecto básico para el uso de una red es el acceso a la información; de acuerdo esta premisa básica, una red que ofrece acceso a una base de datos se convierte en un instrumento vital para el desarrollo de las actividades de toda organización. Los servidores de bases de datos constituyen una herramienta fundamental para el almacenamiento de datos de los usuarios, que a diferencia del almacenamiento en directorios, una base de datos ofrece información estructurada, organizada y disponible para el uso en otras aplicaciones.

El correo electrónico es otro servicio que permite mejorar la comunicación entre los usuarios de la organización a un costo menor que el uso de teléfonos. Asimismo, se pueden ofrecer otros servicios que varían dependiendo de las necesidades de las organizaciones, servidores de imágenes, de impresión y videoconferencias son sólo algunos ejemplos de ellos.

Una red de computadoras también puede funcionar como una puerta para acceder a todos los servicios que brinda Internet, entre los que destacan, el correo electrónico, las videoconferencias, el chat, las descargas de programas, los buscadores, compras, noticias y un sinfín de servicios informáticos.

1.5 MODELOS *OSI* Y TCP/IP

Para poder establecer una comunicación entre computadoras, lo mismo que para establecerla entre personas, es necesario contar con una serie de normas que regulen dicho proceso, a este conjunto de normas se le denomina protocolo y es lo que hace posible el intercambio fiable de comunicación entre dos equipos informáticos. Esas normas son definidas por organismos internacionales de normalización.

Al principio del desarrollo de la computación cada fabricante establecía los procedimientos de comunicación entre sus computadoras de forma independiente, por lo que resultaba muy difícil la comunicación entre computadoras de fabricantes distintos. Poco a poco se fue haciendo necesario disponer de unas normas comunes que permitiesen la intercomunicación entre todas las computadoras.

1.5.1 Modelo *OSI*

De todos los protocolos propuestos destaca el modelo *OSI* (Open Systems Interconnection), Interconexión de Sistemas Abiertos, que fue propuesto por la Organización Internacional de Normalización (ISO).

ISO es una organización no gubernamental fundada en 1947, tiene por misión la coordinación del desarrollo y aprobación de estándares a nivel internacional. Su ámbito de trabajo cubre todas las áreas, incluyendo las redes locales, a excepción de las áreas electrotécnicas que son coordinadas por IEC (International Electrotechnical Commission).

Cada país únicamente puede estar representado en ISO por una organización y en el caso de Estados Unidos está representada por ANSI (American National Standards Institute).

El modelo *OSI*, cuya actividad se empezó a desarrollar en 1977 y llegó a constituirse como estándar internacional en 1983, trata de establecer las bases para la definición de protocolos de comunicación entre sistemas informáticos. Este modelo propone dividir en niveles todas las tareas que se llevan a cabo en una comunicación entre computadoras, todos los niveles están bien definidos y no interfieren con los demás.

En total se tienen siete niveles, como se aprecia en la figura 1.5, Modelo *OSI*, los cuatro primero tienen funciones de comunicación y los tres restantes de proceso. Cada uno de los siete niveles dispone de los protocolos específicos para el control de dicho nivel[2].

Nivel físico

En este nivel se definen las características eléctricas y mecánicas de la red necesarias para establecer y mantener la conexión física, se incluyen las dimensiones físicas de los conectores, los cables y los tipos de señales que van a circular por ellos. Los sistemas de redes locales más habituales definidos en este nivel son: Ethernet, red en anillo con paso de testigo (*Token ring*) e interfaz de datos distribuidos por fibra (*FDDI*, Fiber Distributed Data Interface).

Nivel de enlace de datos

Se encarga de establecer y mantener el flujo de datos que discurre entre los usuarios. Controla si se van a producir errores y los corrige (se incluye el formato de los bloques de datos, los códigos de dirección, el orden de los datos transmitidos, la detección y la recuperación de errores). Las normas Ethernet y *Token ring* también están definidas en este nivel.

Nivel de red

Se encarga de decidir por dónde se han de transmitir los datos dentro de la red, incluye la administración y gestión de los datos, la emisión de mensajes y la regulación del tráfico de la red. Entre los protocolos más utilizados definidos en este nivel se encuentran: Protocolo Internet (*IP*) y el intercambio de paquetes entre redes (IPX, Internetwork Packet Exchange) de Novell.

Nivel de transporte

Asegura la transferencia de la información a pesar de los fallos que pudieran ocurrir en los niveles anteriores, incluye la detección de bloqueos, caídas del sistema, asegurar la igualdad entre la velocidad de transmisión y la velocidad de recepción y la búsqueda de rutas alternativas. Entre los protocolos de este nivel más utilizados se encuentran el Protocolo de Control de Transmisión (*TCP*, Transmission Control Protocol) de Internet y el intercambio secuencial de paquetes (SPX, Sequenced Packet Exchange) de Novell.

Nivel de sesión

Organiza las funciones que permiten que dos usuarios se comuniquen a través de la red. En este nivel se consideran las tareas de seguridad, contraseñas de usuarios y la administración del sistema.

Nivel de presentación

Traduce la información del formato de la máquina aun formato comprensible por los usuarios, se incluye el control de las impresoras, emulación de terminal y los sistemas de codificación.

Nivel de aplicación

Se encarga del intercambio de información entre los usuarios y el sistema operativo, se incluye la transferencia de archivos y los programas de aplicación como telnet, ftp, messenger, correo electrónico, etc.

Figura 1.5, Modelo OSI

El proceso que se produce desde que un usuario envía un mensaje hasta que llega a su destino consiste en una bajada a través de todos los niveles, con sus correspondientes protocolos, desde el nivel séptimo hasta llegar al primero. Allí se encontrará en el canal de

datos que le dirigirá al usuario destino y volverá a subir por todos los niveles hasta llegar al último de ellos, la figura 1.6, Niveles del Modelo *OSI*, muestra este proceso.

Los niveles inferiores proporcionan servicios a los niveles superiores; cada nivel dispone de un conjunto de servicios, que están definidos mediante protocolos. Los programadores y diseñadores de productos sólo deben preocuparse por los protocolos del nivel en el que trabajan, los servicios proporcionados a los niveles superiores y los servicios proporcionados por los niveles inferiores.

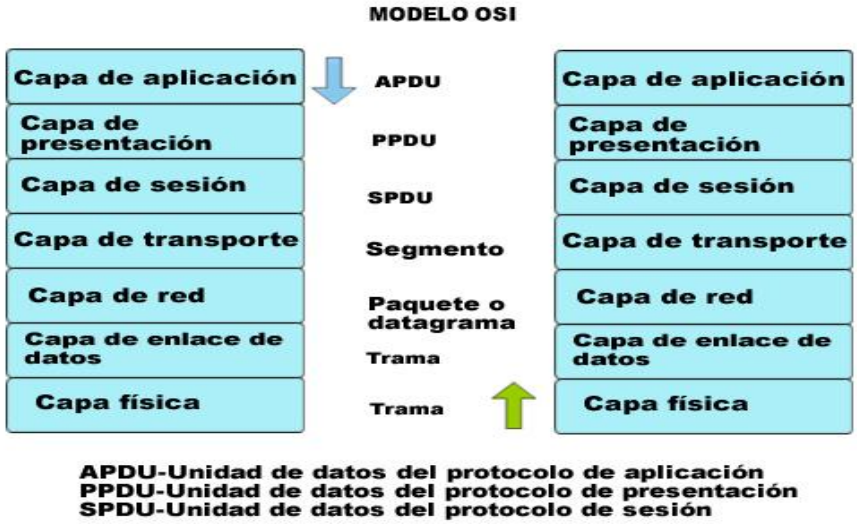

Figura 1.6, Niveles del Modelo OSI

1.5.2 Modelo TCP/IP

TCP/IP fue desarrollado mucho antes de que el modelo OSI de siete capas propuesto por ISO fuera especificado y encaja en su propio modelo de cuatro capas. Todos los protocolos del conjunto de protocolos TCP/IP se ubican en las tres capas superiores de este modelo.

El protocolo TCP/IP es en realidad un conjunto de protocolos aceptados por la industria que permiten la comunicación en un entorno muy variado[3]. Permite acceder a Internet y a sus recursos y se ha convertido en un estándar en la interconexión de redes y en la interoperabilidad entre distintos tipos de equipos. Fue desarrollado por el departamento de la defensa de Estados Unidos con el objetivo de mantener enlaces de comunicación entre sitios en el caso de una guerra mundial; utiliza una arquitectura escalable, cliente-servidor adaptable a las necesidades de conexión.

Este modelo consta de cuatro niveles, como se muestra en la figura 1.7, Modelo TCP/IP, el más bajo es la capa de interfaz de red, que especifica los detalles físicos relativos a la

forma de transmisión de los datos por una red, en este nivel se define el medio eléctrico de transmisión de los datos entre dispositivos de hardware como cable coaxial, fibra óptica o cable par trenzado. En esta capa se implementan los estándares Ethernet, *Token ring*, *FDDI*, X.25, *Frame relay* y otros similares.

La capa de Internet empaqueta los datos en datagramas *IP*, que contienen información de las direcciones de origen y destino utilizada para reenviar los datagramas entre hosts y a través de redes, además, realiza el enrutamiento de los datagramas *IP*. Entre los protocolos incluidos están *IP*, *ICMP*, *IGMP* y *ARP*.

La capa de transporte permite administrar las sesiones de comunicación entre equipos host. Define el nivel de servicio y el estado de la conexión utilizada al transportar datos. Entre los protocolos incluidos están *UDP* y *TCP*.

La capa de aplicación define los protocolos de aplicación TCP/IP y cómo se conectan los programas de host a los servicios del nivel de transporte para utilizar la red. La capa de aplicación contiene protocolos y servicios de aplicación tales como HTTP, Telnet, FTP, DNS, SMTP y X Windows.

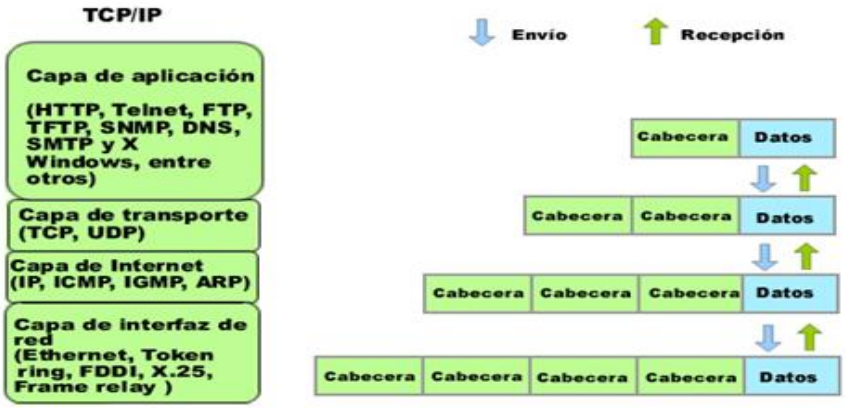

Figura 1.7, Modelo TCP/IP

De igual manera que en el modelo *OSI*, el proceso que se produce desde que un usuario envía un mensaje hasta que llega a su destino consiste en una bajada a través de todas las capas del modelo TCP/IP, con sus correspondientes protocolos, desde la capa de aplicación hasta la capa de interfaz de red. Allí se encontrará en el canal de datos que le dirigirá al usuario destino y volverá a subir por todas las capas hasta llegar nuevamente a la capa de aplicación.

Se puede establecer una correspondencia entre los modelos *OSI* y TCP/IP como se muestra en la figura 1.8, Modelos *OSI* y TCP/IP.

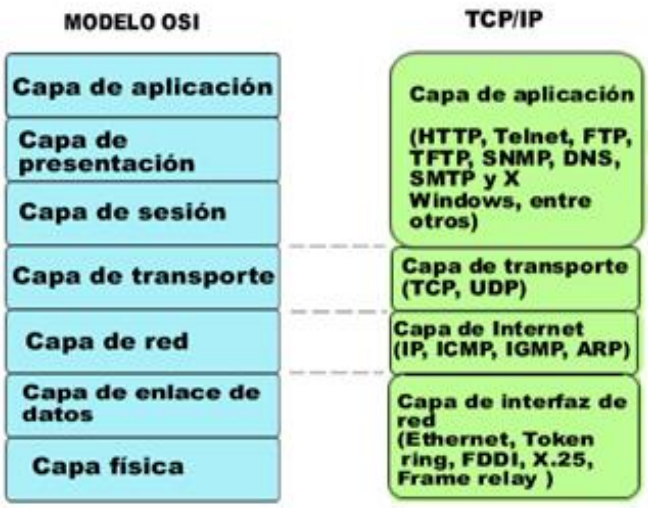

Figura 1.8, Modelos OSI y TCP/IP

1.6 PROTOCOLOS DE RED

Un protocolo es una especificación estándar para dar formato a los datos y transferirlos de una manera que tanto los equipos emisores como los receptores sigan las mismas reglas y así asegurar la comunicación y entendimiento entre ambos.

A continuación se analizan los protocolos básicos, cabe mencionar que esta enumeración no es exhaustiva y por lo tanto no se considera la totalidad de protocolos y subprotocolos existentes.

1.6.1 Protocolo Internet

El protocolo Internet (*IP*, Internet Protocol) se encarga de la clasificación y entrega de paquetes o datagramas en la capa de red del modelo *OSI* y su equivalente *IP* en el modelo TCP/IP. Cada paquete entrante o saliente, o datagrama *IP*, incluye la dirección *IP* origen del remitente y la dirección *IP* del destinatario deseado; las direcciones *IP* de un datagrama no se modifican durante la transmisión de un paquete por la red. *IP* es el responsable del enrutamiento de los paquetes; los datagramas pasan a *IP* desde el *UDP* y *TCP* y desde los adaptadores, *IP* examina la dirección de destino de cada datagrama, la compara con una tabla de enrutamiento y decide qué acción adoptar. Las rutas y las tablas de enrutamiento se pueden configurar de forma estática o dinámica mediante el protocolo de enrutamiento de Internet o abrir la ruta de acceso más corta primero (OSPF, Open Shortest Path First).

El protocolo *IP* es de los denominados “no fiable”, ya que, en busca de una mayor eficiencia, no realiza comprobaciones de una correcta recepción por parte del equipo destino. Para el control de los posibles errores, existe el protocolo *ICMP* que genera los

avisos pertinentes, además la fiabilidad de la comunicación la proporcionan protocolos superiores como *TCP*. En la figura 1.9, Cabecera *IP*, se muestran los campos que conforman una cabecera *IP*.

Figura 1.9, Cabecera *IP*.

Versión (version) (4 bits): el campo versión indica el formato de la cabecera *IP*, en la actualidad sólo se manejan dos:

- IPV4: IP estándar
- IPV6: La versión 6 del protocolo.

Longitud (Internet header length, IHL) (4 bits): indica la longitud de la cabecera expresada en palabras de 32 bits, siendo la longitud mínima 5 palabras, que equivaldría a un paquete *IP* con una cabecera sin opciones.

Tipo de servicio (type of service, TOS) (8 bits): el tipo de servicio se emplea para determinar parámetros como la fiabilidad, la procedencia, el retardo y la capacidad de salida asociadas a este paquete.

Longitud total del datagrama (total length) (16 bits): especifica la longitud total del datagrama *IP* expresada en bytes, incluyendo la cabecera y todos los encapsulados de ésta.

Identificación (identification) (16 bits): este campo identifica de manera unívoca cada paquete enviado por el emisor, de esta forma, se hace posible la reconstrucción, por parte del receptor, de paquetes grandes que tuvieron que ser fragmentados en algún punto de la red.

Banderas (flags) (3 bits):

- Flag more flag (MF): es el primer bit y se usa en la fragmentación, indica si es el último paquete o si le siguen más.

- Don´t fragment (DF): es el segundo bit y especifica si el emisor permite una fragmentación del datagrama.
- El tercer bit en la actualidad no se usa.

Desplazamiento del fragmento (fragment offset) (13 bits): este fragmento indica la posición (desplazamiento) del paquete dentro del paquete original, lo que hace posible su reconstrucción en el destino.

Tiempo de vida (time to live, TTL) (8 bits): indica el tiempo máximo que el paquete permanecerá en la red; si el valor llega a 0, el paquete será destruido, este valor realmente indica el número de saltos (router) por los que puede llegar a pasar.

Protocolo (protocol) (8 bits): especifica el protocolo del siguiente nivel que recibirá los datos, el contenido del campo comenzará, por ejemplo, por *TCP* o *UDP*.

Suma de comprobación (header checksum) (16 bits): este campo se usa par comprobar la integridad de la cabecera, esto es debido a que, durante la transmisión, ciertos campos de la cabecera van modificando su tamaño en cada salto; por ejemplo, se ha de recalcular este valor y verificarse en cada punto intermedio para garantizar una transmisión correcta.

Dirección *IP* de origen (source address) (32 bits): indica la dirección *IP* del dispositivo que originó la transmisión.

Dirección *IP* Destino (destination address) (32 bits): indica la dirección *IP* del dispositivo de destino.

Opciones (options): de longitud variable, puede tener 0 o más opciones, guarda las opciones solicitadas por el emisor, generalmente de seguridad, source routing, timestamps, etc.

Relleno (padding): de longitud variable, su tarea es asegurar que la cabecera *IP* acaba en múltiplo de 32 bits.

Datos (data): de longitud variable, contiene los datos a enviar, siendo su longitud múltiplo de 8 bits y el tamaño máximo 65,535 bytes (64 Kbytes). El campo comenzará con el contenido de la cabecera del protocolo del siguiente nivel, *TCP* o *UDP*.

Para mayor información consulte el RFC 791 - http://www.faqs.org/rfcs/rfc791.html

1.6.2 Protocolo *TCP*

El protocolo de control de transmisión (*TCP*, Transport Control Protocol) proporciona un servicio seguro, basado en conexión de flujo de bytes a las aplicaciones. Se utiliza para inicios de sesiones, para compartir archivos e impresoras, para procesos de duplicación entre controladores de dominio, para la transferencia de listas exploradas y otras funciones comunes. Se puede utilizar únicamente para comunicaciones uno a uno en la que el punto de partida y los extremos están definidos y la ruta de transmisión se establece mediante un protocolo de enlace.

El protocolo *TCP* es empleado por la mayoría de los servicios dentro de la red, es un protocolo de los considerados fiables, desde el punto de vista de que asegura una correcta recepción de los paquetes, ya que utiliza secuencias y confirmación de recepciones mediante el uso de unos paquetes especiales denominados ACK (ACKnowledgement). Es un protocolo, en consecuencia orientado a la conexión. Esto quiere decir que ambos interlocutores han de conocer sus direcciones *IP* para establecer la conexión y emitir y recibir por un puerto especificado en el paquete y conocido por ambos, emisor y receptor, de igual modo que la conexión, se ha de finalizar de una manera correcta. El protocolo *TCP* establece una conexión de tipo stream, es decir, de flujos de información de 8 bits (1 byte) por lo que la información viaja sin marcadores de datos. En la siguiente figura, 1.10, Protocolo *TCP*, se muestran los campos de esta cabecera.

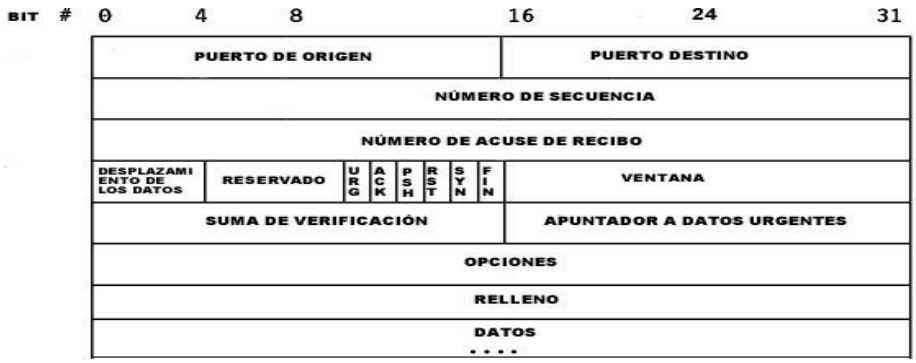

Figura 1.10, Protocolo TCP.

Puerto de origen (source port) (16 bits): puerto de origen o aplicación en el sistema origen.

Puerto destino (destination port) (16 bits): puerto de destino o aplicación en el sistema destino.

Número de secuencia (sequence number) (32 bits): indica el número de secuencia del primer byte de datos del segmento *TCP*. Cuando el flag SYN está activo (establecimiento

de conexión) éste se considera como parte de los datos y se usa el número inicial de la secuencia.

Número de acuse de recibo (acknowledgement number) (32 bits): si el flag ACK está activo, este campo contiene el valor del siguiente número de secuencia que el sistema espera recibir. Este campo siempre tendrá un valor una vez que la conexión se haya establecido.

Desplazamiento de los datos (data offset) (4 bits): número de palabras de 32 bits en la cabecera *TCP*, de tal manera que queda totalmente claro dónde finaliza la cabecera y comienzan los datos.

Reservado (reserved) (6 bits): está reservado para un uso futuro.

Banderas (flags) (6 bits): determinan el funcionamiento de la conexión *TCP*.

URG: flag de urgencia, especifica al receptor la existencia de información urgente dentro del flujo de datos.

ACK: reconoce una recepción correcta de datos, indicando que el campo "acknowledgement number" es válido.

PSH: la función push indica al receptor que ha de pasar los datos a la capa superior (aplicación) tan rápido como sea posible.

RST: indica un reinicio en la conexión.

SYN: paquete que se usa durante el inicio de la sesión para sincronizar los número de secuencia.

FIN: indica que no existen más datos y comienza la negociación de fin de comunicación.

Ventana (window) (16 bits): la ventana *TCP* se utiliza para el control del flujo de datos. Contiene el número de bytes de datos que pueden ser recibidos, empezando por el que se indica en el campo acknowledgement.

Suma de verificación (checksum) (16 bits): es un campo para el control de la integridad de la cabecera *TCP*. En este caso el valor 0 no es un valor correcto.

Apuntador a datos urgentes (urgent pointer) (16 bits): el objetivo de este campo es indicar el puntero con el desplazamiento donde comienzan los datos urgentes, con la finalidad de que el receptor pueda adquirirlos directamente. Este campo sólo tiene sentido cuando el flag URG está activo.

Opciones (options): de longitud variable, puede tener 0 o más opciones, lo más típico dentro de *TCP* es MSS (Maximum Segment Size) permitiendo al sistema especificar el tamaño máximo del segmento que espera recibir, suele indicarse en el paquete con flag SYN, durante el establecimiento de la comunicación.

Relleno (padding): de longitud variable, tiene como misión hacer que la cabecera tenga un tamaño múltiplo de 32, añadiendo ceros para conseguirlo.

Datos (data): este campo almacena los datos y, evidentemente, es de longitud variable.

Para mayor detalle RFC 793 Consulte: http://www.faqs.org/rfcs/rfc793.html

1.6.3 Protocolo de mensajes de control de Internet

El protocolo de mensajes de control de Internet (*ICMP*, Internet Control Messaging Protocol) proporciona funciones de mantenimiento y enrutamiento para el Protocolo de Internet. Los mensajes *ICMP* se encapsulan en datagramas *IP* y pueden enrutarse entre redes. El protocolo genera y mantiene tablas de enrutamiento, realiza labores de descubrimiento de enrutadores, ayuda a determinar la unidad máxima de transmisión de ruta (PMTU, Path Maximum Transmission Unit), ajusta el flujo de control para evitar la saturación de enlaces o enrutamiento, y proporciona herramientas de diagnóstico, como ping.

ICMP se emplea para el control de errores, es decir, para la notificación de errores o para ciertas situaciones que requieran determinada atención. El protocolo se encuentra definido en el RFC 792.

Los paquetes *ICMP* viajan dentro de paquetes *IP* y este protocolo es, en ocasiones, considerado de un nivel superior. Existen diversos tipos de mensajes para la notificación de errores, así como peticiones de información. Cuando se realiza un ping entre dos máquinas se están enviando paquetes empleando este protocolo.

Un paquete *ICMP* contiene 8 primeros bits del paquete *IP* que lo generó, así el sistema receptor del paquete será capaz de extraerlo de la red y sabrá asociarlo a *TCP* o a *UDP*. En la figura 1.1, Protocolo *ICMP*, se muestra el contenido de este tipo de paquete.

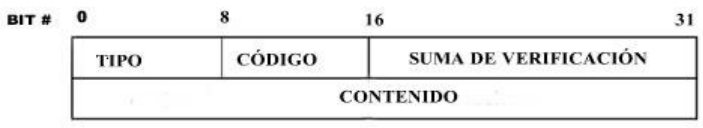

Figura 1.11, Protocolo *ICMP*.

Tipo (type) (8 bits): identifica el tipo específico de mensaje *ICMP*, puede tener 15 posibles valores.

Código (code) (8 bits): para un mismo tipo de mensaje, en este campo se especifican las diferentes condiciones.

Suma de verificación (checksum) (16 bits): es un campo de comprobación de integridad para el total del mensaje *ICMP*, este campo es de carácter obligatorio.

Contenido (contents): de longitud variable, depende del tipo de mensaje.

Si se analizan los mensajes *ICMP* de error, se puede apreciar que uno de los más típicos es el de tipo 3, el que no se puede llegar al destino (destination unreachable).

Además este protocolo se usa para la obtención de información de la red mediante la utilización de pares de paquetes (petición/respuesta); es curioso también su empleo para la obtención de la fecha y hora mediante el manejo de timestamps.

Además el uso más típico es para conectividad, como ya se mencionó antes, utilizando el conocido ping.

1.6.4 Protocolo de resolución de direcciones

El protocolo de resolución de direcciones (*ARP*, Address Resolution Protocol) resuelve direcciones *IP* para las direcciones de control de acceso a medios (MAC) utilizadas por los dispositivos hardware de red, como adaptadores. Una dirección MAC identifica un dispositivo dentro de su propia red física mediante un número de 6 bytes (48 bits) programados en la memoria de sólo lectura. Las direcciones MAC suelen visualizarse en notación hexadecimal.

El funcionamiento de este protocolo está descrito en el RFC 826. El protocolo *ARP* tiene como misión realizar la asociación entre la dirección *IP* y un dispositivo físico, es decir, la dirección física (MAC). Ejemplo: Dirección *IP* 192.168.0.51, Dirección MAC 00-10-60-5b-13-9c.

En la figura 1.12, Protocolo *ARP*, se muestran los campos contenidos en esta cabecera.

Figura 1.12, Protocolo *ARP*.

Descripción de los campos de ARP

Protocolo de hardware (hardware protocol) (16 bits): tecnología de red empleada por debajo de TCP/IP (Ethernet).

Protocolo de red (network protocol) (16 bits): tipo de protocolo empleado a nivel 3 (nivel *IP*).

Longitud de dirección de red de hardware (hardware address length) (8 bits): longitud de la dirección de red de hardware.

Longitud de la dirección de red (network address length) (8 bits): longitud de la dirección de red *IP*.

Operación (operation) (16 bits): tipo de operación que nos da información sobre si se trata de una petición o una respuesta *ARP*.

Dirección física de la interfaz del emisor (sender hardware address) (48 bits): dirección física (MAC) de la interfaz de red del emisor.

Dirección *IP* del emisor (sender network address) (32 bits): dirección *IP* del emisor.

Dirección física de la interfaz del receptor (target hardware address) (48 bits): dirección física (MAC) de la interfaz de red del receptor.

Dirección *IP* del receptor (target network address) (32 bits): la dirección *IP* del receptor.

La conexión se realiza a nivel físico, pero existe algún punto en la red donde tiene que haber una asociación entre esas direcciones físicas, es decir, las MAC de los dispositivos y las direcciones *IP*. Los encargados de efectuar este tipo de relaciones serán los routers y los switches que son capaces de guardar información a ese nivel. Si bien dichos dispositivos memorizan la tabla de correspondencias entre la dirección *IP* de una máquina y su

dirección MAC, para conseguir el encapsulamiento entre las capas, existe un protocolo que en esencia tiene una misión, éste protocolo es el *ARP* o protocolo de resolución de nombre.

Cuando un equipo intenta establecer una conexión con una dirección *IP* determinada, necesita determinar cuál es la MAC del dispositivo de red al que le corresponde la dirección *IP* que está solicitando; mediante la dirección *IP*, los paquetes van saltando de router en router hasta que llegan por fin a la red donde se encuentra dicha dirección *IP* y, una vez allí, es el router el que determina la asociación con la MAC y se establece la conexión.

El router tiene memorizadas esas correspondencias en una tabla, la cual no es infinita, y las relaciones se van generando dinámicamente según su uso, por ello las relaciones antiguas serán eliminadas de la tabla dando paso a las nuevas. La manera por la cual el router conoce estas asociaciones es mediante peticiones *ARP*, en las cuales pregunta quién tiene una determinada MAC y el equipo que coincida responderá dando su dirección *IP* y esta será almacenada en las tablas.

La técnica de memorizar esa tabla de resolución aporta velocidad a la red y una reducción drástica del número de paquetes *ARP* que estarían circulando por la red, optimizando el tráfico de éstos, ya que sólo se realiza una petición *ARP* cuando su correspondencia no está dentro de la tabla.

Los routers son piezas fundamentales en toda red, por tanto, su seguridad es considerada como vital para el funcionamiento de la red.

1.6.5 Protocolo de datagrama de usuarios

El protocolo de datagrama de usuarios (*UDP*, User Datagram Protocol) proporciona un servicio de transporte sin conexión, no confiable y utilizado, por lo general, para comunicaciones de uno a muchos, que emplean datagramas *IP* de difusión y multidifusión. Dado que la entrega de los datagramas *UDP* no es segura, las aplicaciones que utilizan *UDP* pueden suministrar sus propios mecanismos de fiabilidad. *UDP* se puede usar para proceso de inicio de sesión, exploración y resolución de nombres.

El protocolo *UDP* es muy simple, no fiable y no orientado a la conexión, por lo que los protocolos de niveles superiores son los encargados de asegurar una transmisión correcta de datos. No orientado a la conexión significa que no espera a recibir confirmación positiva del destinatario de que la transmisión está siendo recibida de manera correcta, se limita a localizar al destinatario y a entregar. Cada operación de envío genera un único paquete de datagrama *UDP*. Se usa en aplicaciones multimedia, para el envío de flujos de información

sin un costo de conexión asociado. La figura 1.13, Protocolo *UDP*, nos muestra el contenido de este datagrama.

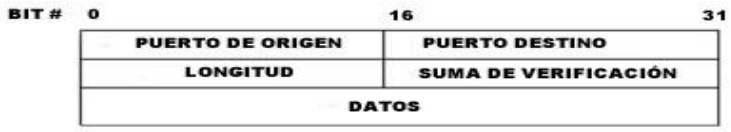

Figura 1.13, Protocolo UDP.

Puerto de origen (source port) (16 bits): puerto de origen o aplicación en el sistema de origen.

Puerto destino (destination port) (16 bits): puerto de destino o aplicación en el sistema destino.

Longitud (length) (16 bits): longitud en bytes del datagrama, incluyendo la cabecera *UDP* y los datos.

Suma de verificación (checksum) (16 bits): comprobación de la integridad de la cabecera *UDP*; este campo puede tomar el valor 0, lo que quiere decir que no se han de realizar comprobaciones de integridad.

Datos: contenido de longitud variable.

Para mayor detalle RFC 793 Consulte: http://www.faqs.org/rfcs/rfc768.html

1.7 NORMAS DE SEGURIDAD

Debido a la necesidad de contar con un estándar de medición de la seguridad en los sistemas informáticos, se han establecido una serie de normas de seguridad o criterios con los que se puede evaluar el grado de confianza que se puede depositar en un sistema. Ejemplos de estas normas de seguridad son los TCSEC (Trusted Computer System Evaluation Criteria o Libro Naranja) del Departamento de Defensa de los Estados Unidos, los criterios de la evaluación de seguridad de la tecnología de información (ITSEC), un estándar europeo, y los criterios comunes para evaluación de seguridad de tecnología de la información (Common Criteria for Information Technology Security - CCIT-SE)[5].

Criterios comunes para evaluación de seguridad de tecnología de la información

Los criterios comunes son una norma internacional para evaluar la seguridad de los productos de tecnología de la información basados en los criterios europeos,

norteamericanos y canadienses existentes para la evaluación de la seguridad de tecnología de la información[6].

Los criterios comunes se originaron con proyectos cooperativos que estaban relacionados con la Organización Internacional de Estándares (ISO). En 1999 fue adoptada por ISO como Tecnología de información –Técnicas de seguridad- Criterios de evaluación para la seguridad de tecnología de la información (ISO/IEC 15408).

La certificación de un producto sobre los criterios comunes se debe hacer por una entidad independiente aprobada. El proceso de evaluación certifica que un producto verifica los aspectos siguientes:

- Los requerimientos del producto están definidos correctamente.
- Los requerimientos están implementados correctamente.
- El proceso de desarrollo y documentación del producto cumple con ciertos requerimientos.

Los Criterios Comunes están formados por tres partes: introducción y modelo general, requerimientos de seguridad funcional y requerimientos de garantía de seguridad.

La introducción y modelo general describen el formato de los criterios comunes y como fueron establecidos, la segunda parte, requerimientos de seguridad funcional define once clases denominadas clases de requerimientos funcionales y cada una de ellas cubre un área particular de la seguridad, a su vez, cada clase cuenta con una o más familias funcionales que tratan aspectos específicos de la totalidad de la clase en cuestión. Las clases antes mencionadas son:

- Auditoría de seguridad
- Comunicación
- Soporte de cifrado
- Protección de datos de usuario
- Identificación y autenticación
- Administración de la seguridad
- Privacía
- Protección de las funciones de seguridad del objeto de evaluación
- Utilización de recursos
- Control de acceso
- Caminos/canales confiables

La tercera parte de los criterios define a las clases de garantía de seguridad y los niveles de garantía. Las clases de garantía son aquellos requerimientos necesarios para garantizar la seguridad que brinda el objeto de evaluación. Estos requerimientos están agrupados en siete

clases y a su vez, cada clase está formada por una o más familias de garantías. Las siete clases de garantía de la seguridad son:

- Administración de la configuración
- Distribución y operación
- Desarrollo
- Documentos guía
- Soporte al ciclo de vida
- Pruebas
- Vulnerabilidad de los bienes

Los niveles de garantía o EAL (Evaluation Assurance Level) definen una escala para garantizar la medición y los criterios para la evaluación de perfiles de protección. Los niveles de garantía son siete van desde el nivel de seguridad más bajo (EAL_1) hasta el más alto (EAL_7):

EAL_1 Funcionalidad probada.
EAL_2 Estructuralmente probado
EAL_3 Probado y verificado metódicamente
EAL_4 Diseñado, probado y revisado metódicamente
EAL_5 Diseñado y probado semiformalmente
EAL_6 Diseño verificado y probado semiformalmente
EAL_7 Diseño verificado y probado formalmente

Los criterios comunes proporcionan ventajas a los usuarios de productos de tecnología de la información como son, tener un medio de comparación para varios productos de diferentes desarrolladores; contar con un esquema que permita describir sus necesidades de seguridad y utilizar productos evaluados por una entidad externa de acuerdo con criterios internacionales reconocidos.

En el caso de los desarrolladores, los criterios comunes brindan un medio para comprobar que su producto es adecuado, está bien diseñado y es seguro, y esto constituye las ventajas competitivas que busca todo desarrollador para posicionar su producto en el mercado.

1.8 TIPOS DE AMENAZAS EN UNA RED

Una amenaza se representa a través de una persona, una circunstancia o evento, un fenómeno o una idea maliciosa las cuales pueden provocar daño cuando existe una violación de la seguridad, así, una amenaza es todo aquello que intenta o pretende destruir[6].

En la actualidad las redes de computadoras están expuestas a múltiples amenazas que atentan contra la seguridad de la información y de los recursos de la red. Existen amenazas

relacionadas con el hardware, con el software, catástrofes naturales, fallas humanas o ataques malintencionados. Las amenazas por catástrofes naturales son inherentes a toda organización, sin embargo se puede disminuir el riesgo por este tipo de amenazas mediante una adecuada planeación de las instalaciones y una correcta protección de los equipos.

Las amenazas humanas son, generalmente debidas a errores u omisiones, o al desconocimiento de las personas que utilizan los recursos de la red. Existe otro tipo de amenaza que se diferencia del resto, básicamente porque no se aprovecha de debilidades y vulnerabilidades propias de un componente informático para la obtención de información y es, la amenaza del tipo ingeniería social, que consiste en utilizar artilugios, tretas y otras técnicas que engañan a las personas para que de manera no intencional revelen información de interés para el atacante. También se encuentra dentro de este tipo de amenazas, el fraude, el robo, el sabotaje y el espionaje.

Las amenazas debidas al hardware son, esencialmente debidas a errores de fabricación y al mal uso de los equipos por parte del personal encargado. Las amenazas al software por otra parte, consideran elementos como la mala configuración de los programas, fallas en las aplicaciones, códigos maliciosos, virus y exploits que pueden aprovechar las vulnerabilidades en el software.

Las amenazas a la red pueden ser debidas a la mala configuración de la misma, la implementación de un sistema operativo inadecuado o con una mala administración, controles de acceso inadecuados o deficientes. Tambien se puede hacer una clasificación de amenazas dependiendo de la zona a la que pueda dirigirse un ataque. De esta manera se tendrían tres tipos de amenazas a valorar: amenazas de red, amenazas al servidor y amenazas de aplicaciones, aunque algo muy importante a tomar en cuenta es que estas tres amenazas están íntimamente ligadas y un ataque puede estar dirigido a las tres zonas de vulnerabilidad posibles.

Dentro de las amenazas de red se encuentran todos los ataques efectuados desde fuera de la red que tienen que ver con el tráfico que pasa a través de los routers, switches y a través del *firewall*. Este tipo de ataques en su mayoría se relacionan con la escucha secreta o *eavesdropping* y tiene múltiples variantes. El sniffing, o captura de información que circula por la red mediante herramientas diseñadas con tal fin es un ejemplo de este tipo de amenazas, el desbordamiento de la tabla de direcciones de un switch para bloquear su capacidad de direccionar paquetes a su destino es otro ejemplo muy significativo de las amenazas de red. *IP spoofing* o *MAC Address spoofing*, método en el que el atacante modifica la dirección *IP* o la dirección MAC de origen de los paquetes de información para

hacerse pasar por un usuario válido también se encuentra dentro de las amenazas dirigidas a explotar vulnerabilidades inherentes al diseño y seguridad de la red.

Las amenazas al servidor representan un peligro constante sobre el sistema operativo y sus elementos de control y administración y en general se caracterizan por explotar todas las vulnerabilidades relacionadas con la configuración del servidor, del software mismo y de los servicios ofrecidos por el servidor. Por ello es muy importante mantener actualizado el software con que trabajamos, aplicar los parches respectivos y tener información constante sobre las vulnerabilidades recientes encontradas en los servicios que se ofrecen en el servidor, a fin prevenir daños graves.

Un aspecto importante es el cuidado de las bases de datos y el correo electrónico, donde los niveles de confidencialidad, integridad y disponibilidad de la información contenida en ambos servicios determinan el nivel de protección y configuración de seguridad. Ataques como la inyección de SQL pueden provocar pérdidas importantes de información en bases de datos y requiere tanto de una buena protección a nivel de aplicación como de una protección a nivel de servidor.

Finalmente los ataques a aplicaciones, como los *exploits*, que buscan explotar las vulnerabilidades del sistema en general; la *denegación de servicios*, ataque en el cuál se busca dejar sin disponibilidad un servicio en la red; el defacement, ataque en el que se modifica el contenido de un sitio web, son ejemplos de este tipo de amenazas dirigidas contra las aplicaciones.

En general los ataques pueden estar dirigidos a más de un objetivo y buscar por todos los medios las vulnerabilidades del sistema. Incluso pueden perpetrarse ataques desde dentro de la red. Este tipo de amenazas puede ser desde el acceso no autorizado a directorios y archivos hasta la manipulación malintencionada de datos, cambios de archivos sin autorización y manejo no autorizado de recursos. Las amenazas de rechazo se asocian con usuarios que niegan haber ejecutado una acción, pero no existe forma de probar lo contrario. Un ejemplo de este tipo de amenaza es cuando un usuario puede realizar una operación prohibida en un sistema que no tiene la capacidad de rastrear dicha operación.

Una vez que el atacante superado de manera eficaz todas las defensas del sistema, es cuando el sistema corre mayores riesgos. La detección de la intrusión es de vital importancia en este punto. Si el atacante ha logrado el ingreso y trata de hacer una

elevación o escalada de privilegios, implementar una *puerta trasera* o utilizar los recursos de la red, es necesario tener un medio de control que permita verificar que acciones extrañas se llevan a cabo en el servidor y que ponen en peligro al sistema. Este es el tipo de amenaza contra las cuáles un sistema de detección de intrusos se enfoca y es el único medio de defensa capaz de detectar, prevenir y llevar a cabo acciones que permitan que dicha intrusión sea lo menos dañina posible al sistema.

Como se puede observar, existen muchas amenazas a las redes, en la tabla 2.1 Tipos de amenazas, se muestra una clasificación de las mismas. Una buena planeación y políticas de seguridad adecuadas son necesarias para tener una red lo más segura posible. Las medidas a implementar comienzan con la seguridad física del equipo, pero en cuanto al software se refiere, se determinan de acuerdo con las necesidades de confidencialidad, integridad y disponibilidad de la información que se requiere para la red, de acuerdo con el nivel de protección y configuración de seguridad que se requiere para los recursos de la red y los servicios que la misma ofrecerá a sus usuarios. Esta determinación de requerimientos es una tarea que requiere un planeación previa del tipo de sistema operativo que se requiere, los servicios a ofrecer, las características de seguridad de la red, del *firewall* y del sistema de detección de intrusos, aspectos que se analizan en el siguiente capítulo.

Tabla 1.2 Tipos de amenazas

Amenazas	Descripción
Humanas	Errores, omisiones, robo, fraude, ingeniería social, intrusos, espionaje.
Red	Sistemas mal diseñados, mala administración, accesos mal configurados.
Software	Fallos en las aplicaciones, códigos maliciosos, virus, exploits.
Hardware	Errores de fabricación, uso inadecuado.
Naturales	Terremotos, inundaciones, incendios, etc.