Ingeniería Técnica de Informática de Sistemas

Universistat Oberta de Catalunya

TRABAJO FINAL DE CARRERA

DISEÑO E IMPLEMENTACIÓN DE UN CAPTURADOR DE PAQUETES DE RED

Andrés López Peñaranda

Enero 2005

CONTENIDOS

ÍNDICE DE FIGURAS

Capitulo 1. Introducción y descripción de contenidos

En la actualidad las redes de comunicaciones son básicamente digitales, y el camino avanzado en los últimos 40 años ha sido muy largo, con un desarrollo tan rápido que ha hecho que cualquier previsión quedara muy atrás antes de tiempo (o que a veces se sobreestimara este desarrollo, como ocurrió con el boom de las empresas llamadas .com alrededor del año 2000, y su posterior desplome). Hoy en día podemos decir que las comunicaciones entre ordenadores e Internet están introduciéndose muy rápidamente en todos los órdenes de la rutina diaria.

En este marco de desarrollo aparece este trabajo de final de carrera de la Ingeniería Técnica de Informática de Sistemas de la UOC. Conforme las redes de ordenadores se hacían más complejas, se necesitaban cada vez más herramientas de resolución de problemas que nos ayudaran a hacer diagnósticos de los problemas que nos podemos encontrar en una red de comunicaciones.

Una de estas herramientas son los capturadores de paquetes. Estas herramientas capturan todos los datos que viajan por la red y permiten un análisis más o menos amplio, dependiendo del capturador. Con esta información sabemos exactamente qué datos están transmitiéndose por la red, cómo están formateados y cómo se han formado los paquetes, de manera que nos permiten averiguar si existe algún problema en la comunicación, y donde puede estar este problema. Esta memoria se centra en la descripción de unos de estos capturadores de paquetes que se ha diseñado y construido para el trabajo final de carrera de la ingeniería técnica de informática de sistemas de la UOC.

En este primer capítulo hemos dado una descripción del producto y del marco donde se encuadra dentro de la informática y las telecomunicaciones actuales.

En el segundo capítulo vamos a describir qué son las redes de ordenadores, cómo surgieron, qué objetivos tienen y cómo se estructuran. Para realizar cualquier tipo de comunicación (ya sea verbal, mímica o entre dos ordenadores) se han de realizar una serie de tareas para que el mensaje llegue al otro extremo y pueda ser entendido. En este capítulo veremos cuáles son estas tareas y cómo, cuándo y dónde se realizan en una red de ordenadores.

En el tercer capítulo se describirán los protocolos que soporta el capturador de paquetes diseñado y cómo son sus cabeceras. En las cabeceras de los protocolos se encuentra la información que hace que el mismo pueda llevar a cabo sus funciones. Por

lo tanto, es la parte más importante del protocolo y la parte que analiza un capturador de paquetes.

En el cuarto capítulo veremos como se ha diseñado el programa que implementa el capturador de paquetes, cual es su esquema de funcionamiento y cuales han sido las decisiones de diseño más importantes.

En el quinto capítulo se ha incluido el manual de usuario del programa. Uno de los objetivos era hacer lo más sencillo posible el programa, y leyendo este capítulo se puede ver que el programa es muy sencillo de usar.

En el sexto capítulo haremos un repaso de los objetivos de este trabajo final de carrera y evaluaremos si se han cumplido o no.

En el séptimo y último capítulo estará especificada toda la bibliografía usada para la confección de este trabajo final de carrera.

Capitulo 2. Introducción a las redes de ordenadores

Una definición de una red de ordenadores podría ser la siguiente: conjunto de ordenadores conectados entre sí, que pueden comunicarse compartiendo datos y recursos sin importar la localización física de los distintos dispositivos. A través de una red se pueden ejecutar procesos en otro ordenador o acceder a sus ficheros, enviar mensajes, compartir programas... Como vemos en esta definición, hay tres partes que se deben cubrir para tener una red de ordenadores, que son las siguientes:

- ✓ Los ordenadores han de estar conectados. Es lo que conocemos como medio físico. Actualmente hay muchas **tecnologías** de conexión, que se pueden dividir en dos grandes campos: tecnologías cableadas (por ejemplo, Ethernet y sus variantes) e inalámbricas (por ejemplo, Ethernet 802.11x, UMTS, GPRS, satélites).
- ✓ Los ordenadores han de comunicarse. Para ello utilizan los mensajes. El origen ha de elaborar el mensaje, darle el formato adecuado, enviarlo. La red ha de encaminar ese mensaje al destino correctamente para que llegue. Y el destino ha de recibir el mensaje, entender el formato que tiene, comprender que quiere decir el mensaje y realizar las acciones oportunas (en muchos casos, elaborar un mensaje de respuesta). De estas tareas se encargan la pila de **protocolos** de la arquitectura de red.
- ✓ La finalidad de la comunicación. Se pueden ejecutar procesos, compartir ficheros y programas… Esta función la realizan las **aplicaciones**.

En este proyecto, dada su naturaleza, nos vamos a centrar en analizar algunos de los protocolos involucrados en la comunicación, que son los más importantes de todos los que se utilizan hoy en las redes de ordenadores.

2.1. Breve historia de las redes de ordenadores

Ya en los principios de la informática, durante los años 60, se empezaron a realizar conexiones entre ordenadores usando procedimientos y protocolos existentes. Estas conexiones eran básicamente punto a punto, a través de línea telefónica. Se usaba el MODEM (MOdulador/DEModulador) para transformar y adaptar las señales eléctricas del ordenador (señales digitales) al canal telefónico y sus características

(canal analógico). A partir de aquí surgieron los sistemas de respuesta automática, y varios terminales muy simples se conectaban directamente a un ordenador central a través de línea telefónica, de tal manera que se ofrecía remotamente una consola igual a la que se tenía si estabas delante del ordenador.

A partir de los años 70 aparecen las primeras redes de ordenadores en el sentido que conocemos actualmente. Se desarrollan las primeras redes de área local (por ejemplo, Xerox desarrolla Ethernet), y las compañías telefónicas desarrollan productos específicos para la conexión de estas LAN. Todo este desarrollo, unido a los procesos de estandarización de ISO y al desarrollo de ARPANET (proyecto militar de interconexión de ordenadores), dio como resultado el nacimiento de una red de interconexión a nivel internacional, Internet, en 1983. En un principio sólo involucraba a redes universitarias de Estados Unidos, pero a lo largo de la década Internet siguió creciendo, casi siempre restringida a centros de investigación. A lo largo de la década de los 90, Internet se fue convirtiendo cada vez más en un producto comercial, sobre todo después del nacimiento del World Wide Web (WWW), popular método de compartir información, en el CERN de Ginebra.

Hoy en día, Internet como red de redes, ha demostrado ser una importante herramienta corporativa y comercial. La búsqueda y compartición de información, la descarga de programas, el acceso a redes corporativas (VPN, Virtual Private Network) y servicios tradicionales, como la venta de productos y la banca, han hecho de Internet un recurso casi temido y odiado por fundamentalistas, pero se ha convertido en imprescindible para muchas de nuestras actividades cotidianas.

2.2. Objetivos y aplicaciones de las redes de ordenadores

El principal objetivo de las redes de ordenadores es el compartir recursos, y hacer que todos los programas, datos y equipos estén disponibles para cualquiera de la red que así lo solicite, sin importar la localización física del recurso y del usuario. Otro objetivo es proporcionar una alta disponibilidad a ciertos sistemas, poniendo fuentes alternativas de suministro que pueden no estar localizadas en el mismo sitio. Si la fuente de información habitual no está disponible, se accede a alguna de sus copias, independientemente de donde esté. Y por último, el ahorro económico también es un objetivo, porque se puede colocar un ordenador muy potente (y caro) como servidor y muchos ordenadores pequeños (y más baratos) para cada una de las personas de la red.

Las aplicaciones tradicionales que se dan sobre una red de ordenadores son la World Wide Web (WWW), File Transfer Protocol (FTP), Telnet, etc… Pero las aplicaciones que comercializan las redes de ordenadores y las hacen atractivas al gran público se pueden agrupar en los siguientes tipos:

- ✓ Acceso a programas remotos: por ejemplo, una empresa crea un programa que simula la evolución de un organismo vivo (por ejemplo, un virus) en un cierto entorno. Esta empresa podrá permitir a sus clientes (probablemente previo pago, otra aplicación interesante de las redes de ordenadores) ejecuten esta simulación con sus propios parámetros.
- ✓ Acceso a bases de datos remotas: por ejemplo, cada vez que accedemos a nuestro banco a ver nuestras cuentas y otros productos que tenemos contratados, no hacemos más que un acceso a las bases de datos de nuestro banco, pero solo a la parte que tenemos acceso, que es la nuestra (aquí entra otro aspecto a tener en cuenta con las redes de ordenadores, que es la seguridad). Otro ejemplo sería las reservas de avión, hoteles, etc…
- ✓ Facilidades de comunicación de valor añadido: como el correo electrónico, televisión, tablones de anuncios electrónicos, reducción de viajes, teletrabajo, etc… Las redes de ordenadores contribuyen a que seamos más independientes de la situación geográfica para mantener nuestra forma de vida.

Como hemos visto, las redes de ordenadores aportan una serie de ventajas que pueden revolucionar la sociedad y sus relaciones (no ha sido la primera vez que un avance tecnológico ha provocado este tipo de cambios). A partir de aquí es la propia sociedad, con un sistema parecido a la selección natural, la que debe decidir si es ésta u otra tecnología la adecuada a sus necesidades. De momento, Internet y las redes de ordenadores es la elegida.

2.3. Arquitectura de las redes de ordenadores

Para atacar el problema de realizar una red de ordenadores se ha seguido el principio de "divide y vencerás". Se especificaron las tareas que se debían realizar y se agruparon en 7 capas. Este modelo de referencia fue establecido por la organización de

estándares ISO, y se le llamó modelo de referencia OSI (Open Systems Interconnection). En la siguiente figura se puede ver este modelo:

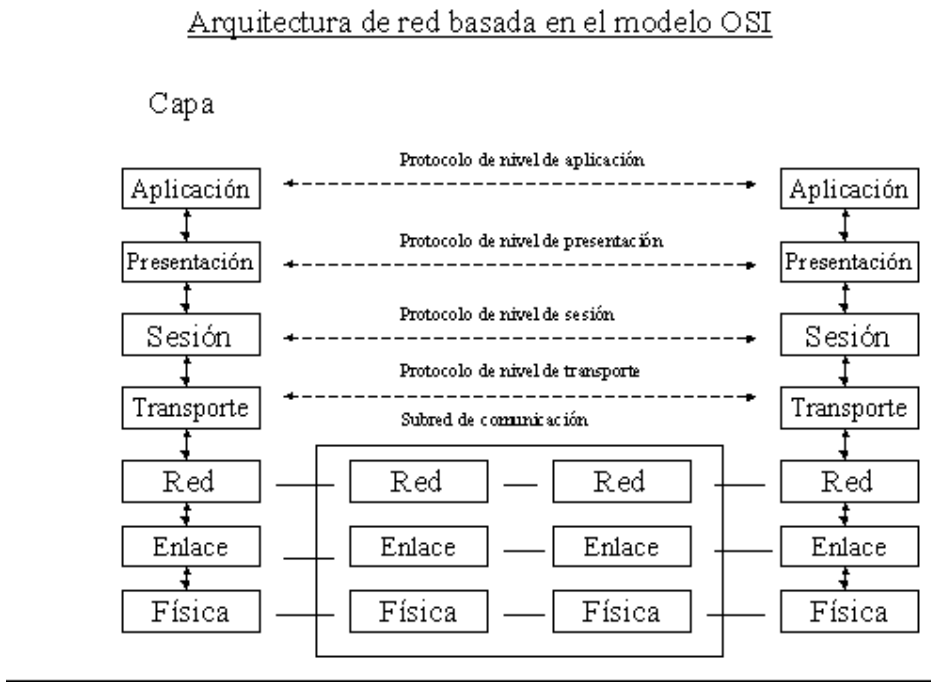

Figura 1. Modelo de Referencia OSI

En cada capa se ejecutan una serie de entidades que realizan las tareas necesarias para que la comunicación sea posible. Las entidades de cada capa sólo se comunican con las entidades de la misma capa usando los protocolos definidos de esa capa. Los nodos de red sólo implementan hasta la capa de red, que es la que implementa el encaminamiento de los paquetes a través de las distintas subredes, mientras que la capa de transporte es la primera capa extremo a extremo. Las tareas definidas para cada capa son las siguientes:

- ✓ **Capa de aplicación**. Relacionada con las funciones de más alto nivel, proporcionando soporte a las aplicaciones o actividades del sistema, suministrando servicios de red a las aplicaciones del usuario y definiendo los protocolos usados por las aplicaciones individuales. Además, sincroniza las aplicaciones que cooperan entre sí y establece acuerdos sobre los procedimientos de recuperación de errores y control de la integridad de los datos. Difiere de las demás capas debido a que no proporciona servicios a ninguna otra capa OSI, sino solamente a aplicaciones que se encuentran fuera del

modelo (procesadores de texto, hojas de cálculo, navegadores web, etc.).

- ✓ **Capa de presentación**. La capa de presentación proporciona sus servicios a la capa de aplicación, garantizando que la información que envía la capa de aplicación de un sistema pueda ser entendida y utilizada por la capa de aplicación de otro, estableciendo el contexto sintáctico del diálogo. Su tarea principal es aislar a las capas inferiores del formato de los datos de las aplicaciones específicas, transformando los formatos particulares (ASCII, EBCDIC, etc.) en un formato común de red, entendible por todos los sistemas y apto para ser enviado por red. También se encarga de los procesos criptográficos necesarios.
- ✓ **Capa de sesión**. La capa de sesión proporciona sus servicios a la capa de presentación, proporcionando el medio necesario para que las entidades de presentación de dos host que se están comunicando por red organicen y sincronicen su diálogo y procedan al intercambio de datos. También se debe restaurar las sesiones sin perdidas de datos en caso de fallo (muy importante en sistemas transaccionales).
- ✓ **Capa de transporte**. La capa de transporte proporciona sus servicios a la capa de sesión, efectuando la transferencia de datos entre dos entidades de sesión. Para ello, divide los datos originados en el host emisor en unidades apropiadas, denominadas **segmentos**, que vuelve a reensamblar en el sistema del host receptor. Mientras que las capas de aplicación, presentación y sesión están relacionadas con aspectos de las aplicaciones de usuario, las tres capas inferiores se encargan del transporte de datos. Además, la capa de transporte es la primera que se comunica directamente con su capa par de destino, ya que la comunicación de las capas anteriores es de tipo máquina a máquina. La capa de transporte intenta suministrar un servicio de transporte de datos que aísle las capas superiores de los detalles del mismo, encargándose de conseguir una transferencia de datos segura y económica y un transporte confiable de datos entre los nodos de la red. Para ello, la capa de transporte establece, mantiene y termina adecuadamente los circuitos virtuales, proporcionando un servicio

confiable mediante el uso de sistemas de detección y recuperación de errores de transporte.

- ✓ **Capa de red**. La capa de red proporciona sus servicios a la capa de transporte, siendo una capa compleja que proporciona conectividad y selección de la mejor ruta para la comunicación entre máquinas que pueden estar ubicadas en redes geográficamente distintas. Es la responsable de las funciones de conmutación y enrutamiento de la información (direccionamiento lógico), proporcionando los procedimientos necesarios para el intercambio de datos entre el origen y el destino, por lo que es necesario que conozca la topología de la red (forma en que están interconectados los nodos), con objeto de determinar la ruta más adecuada. La información se encamina a través de la red en base a las direcciones del paquete de datos, determinando los métodos de conmutación y enrutamiento a través de dispositivos intermedios (routers).
- ✓ **Capa de enlace de datos**. La capa de enlace proporciona sus servicios a la capa de red, suministrando un tránsito de datos confiable a través de un enlace físico. Se ocupa del direccionamiento físico, la topología de red, el acceso a la misma, la notificación de errores, la formación y entrega ordenada de datos y control de flujo. Su principal misión es convertir el medio de transmisión en un medio libre de errores de cualquier tipo. Utiliza códigos de redundancia cíclica (CRC) para detectar y corregir errores, así como acuses de recibo tanto positivos como negativos. También ha de controlar la congestión en la red y el acceso a la red.
- ✓ **Capa física**. La misión principal de esta capa es transmitir bits por un canal de comunicación, de manera que cuanto envíe el emisor llegue sin alteración al receptor. La capa física proporciona sus servicios a la capa de enlace de datos, definiendo las especificaciones eléctricas, mecánicas, de procedimiento y funcionales para activar, mantener y desactivar el enlace físico entre sistemas finales, relacionando la agrupación de circuitos físicos a través de los cuales los bits son transmitidos. Esta capa solamente reconoce bits individuales.

ISO definió protocolos para cada una de las capas, definiendo las reglas de comunicación. Cualquier sistema que implemente los protocolos del modelo OSI podrá comunicarse con todos los sistemas que implementen también los protocolos del modelo OSI.

El modelo de referencia OSI es eso precisamente, un modelo de referencia. Debido a la inercia de ARPANET, núcleo de la actual Internet, se ha implantado de facto la arquitectura TCP/IP en las redes actuales. La filosofía del modelo TCP/IP es similar al modelo OSI, pero en lugar de 7 capas hay solo 4. En la siguiente figura vemos el modelo TCP/IP:

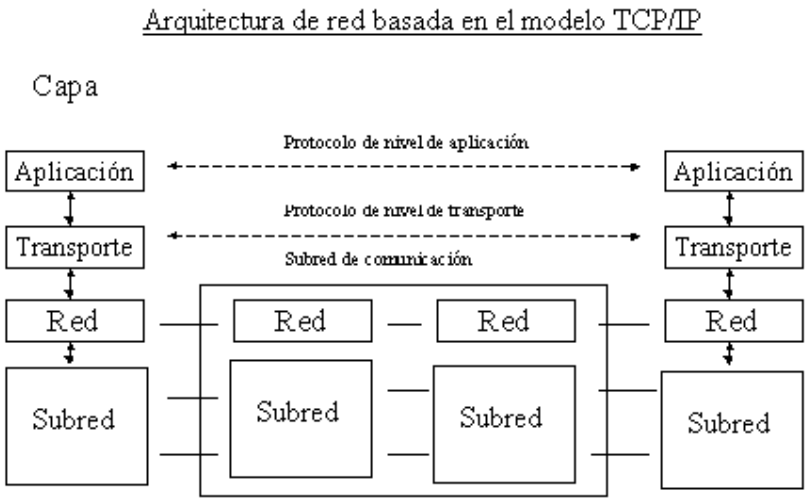

Figura 2. Modelo de referencia TCP/IP

Las funciones de la capa de presentación se suelen integrar en la capa de aplicación, mientras que las funciones de la capa de sesión se incluyen en la capa de transporte. Además se han fusionado dos capas, la de enlace de datos y la física, en una sola capa de subred, para la cual el modelo no especifica nada. Se puede usar los protocolos de enlace de datos y capa física que se quiera.

En este modelo, los protocolos más importantes son el protocolo de red IP (Internet Protocol) y el protocolo de transporte TCP (Transport Control Protocol), que dan nombre al modelo.

En modo de funcionamiento de esta arquitectura se base en la encapsulación: los datos del nivel superior son encapsulados por el protocolo del nivel inferior que añade una cabecera a estos datos para poder realizar sus funciones. En la figura siguiente podemos ver este modo de funcionamiento:

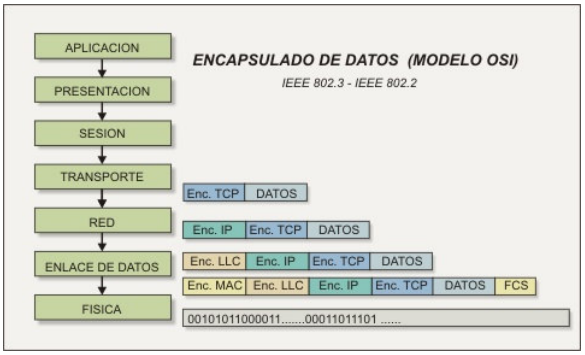

Figura 3. Encapsulado de datos en el modelo de referencia OSI

Un programa capturador de paquetes se centrará en la localización de estas cabeceras añadidas por los protocolos y en su análisis, mostrando los datos que guardan estas cabeceras.

2.4. Resumen

Hemos visto en este capitulo porque son útiles las redes de ordenadores, como se desarrollaron y como se organizan, según los distintos modelos de referencia. Las necesidades de comunicación crecen constantemente, no solo a nivel corporativo o de empresa, sino a nivel particular. Cada vez son más los servicios que se ofrecen a través de Internet (la administración, las empresas, venta por Internet…), y cada vez más gente trabaja desde casa gracias a una VPN (Virtual Private Network) que lo conecta a la red corporativa de su empresa. Las videoconferencias han tenido un gran auge en los últimos años como herramienta de control de costes en las empresas, y están surgiendo nuevas herramientas de trabajo colaborativo que hacen a las empresas cada vez más independientes de su ubicación geográfica. Todo este panorama augura un buen futuro a las redes de comunicaciones.

Capitulo 3. Protocolos de red

Actualmente en Internet, y en las redes de ordenadores en general, se ha impuesto la pila de protocolos TCP/IP, frente a los protocolos del modelo OSI de ISO, aunque se puede ver la similitud entre ambas propuestas, que muchas veces son complementarias. En este capturador nos centramos en redes con arquitectura TCP/IP y como protocolo de enlace de datos Ethernet, ya que ambas tecnologías son estándares de facto.

3.1. Arquitectura TCP/IP

En la siguiente figura se puede ver la arquitectura TCP/IP y algunos de sus protocolos, así como las relaciones entre ellos:

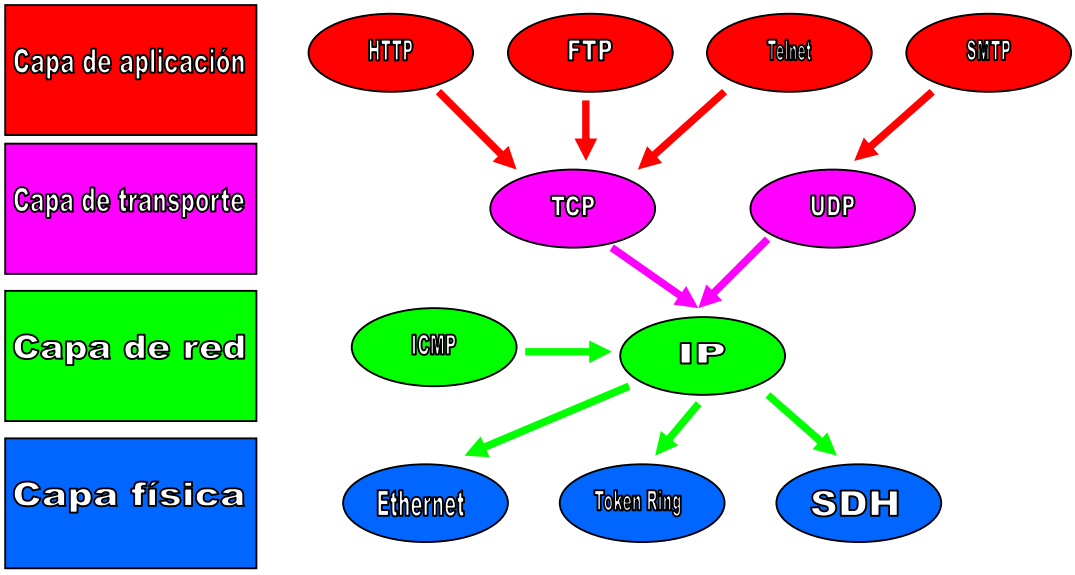

Figura 4. Arquitectura TCP/IP

3.1.1. Capa física

En la capa física tenemos las tecnologías que nos permiten el acceso al medio de comunicación de forma eficiente y adecuada, sea ya el medio guiado (cables o guías de ondas) como medio no guiado (ondas electromagnéticas).

Esta capa agrupa las funciones definidas en los niveles 1 y 2 del modelo OSI. Al ser una capa dependiente del medio físico, se define un protocolo por cada medio físico que se ha de usar en cada momento.

Una de las funciones más importantes de esta capa es la función de control de acceso al medio (MAC, Medium Access Control). Esta capa se encarga que todas las entidades que transmiten a través del medio puedan hacerlo sin transmitir todas al mismo tiempo (colisiones), situación que haría la comunicación imposible.

Un ejemplo de un grupo de protocolos de esta capa es Ethernet. Ethernet define un protocolo para cada medio físico que usa. Algunos de los definidos son los siguientes:

- ✓ Los protocolos 10Base2 y 10Base5 se basan en cable coaxial. Estos protocolos definen una distancia máxima de transmisión, así como número de estaciones máximo y otros parámetros del medio físico (por ejemplo, conectores) y de acceso al medio. La velocidad es de 10 Mbps.
- ✓ Los protocolos 10BaseT, 100BaseT se basan en cables de pares trenzados (apantallados o no), con velocidades de acceso de 10 Mbps y 100 Mbps respectivamente. Los conectores usados son los famosos RJ-45, que son los conectores de red más ampliamente difundidos en la actualidad.
- ✓ El protocolo 100BaseFX está basado en fibra óptica multimodo, y funciona a una velocidad de 100 Mbps.
- ✓ Los protocolos 1000BaseX definen la familia de protocolos Ethernet que funcionan a velocidades de 1000 Mbps, sobre distintos medios (cable de pares trenzado de categoría 5 o 6, fibra monomodo o multimodo).

3.1.2. Capa de red

La capa de red tiene la función principal de encaminar los paquetes a través de la red/redes de forma adecuada de manera que llegue al destino.

El protocolo más importante a este nivel, y que da nombre a la arquitectura es el protocolo IP (Internet Protocol). Sus características principales son:

- ✓ Es no orientado a conexión.
- ✓ La entrega no es fiable, aunque se hará todo lo posible para que el paquete llegue correctamente (filosofía "best effort"). Los protocolos de capas superiores deben de tener mecanismos (si lo consideran oportuno) para recuperar fallos en la entrega.
- ✓ Decide la ruta que ha de seguir un paquete desde el origen al destino.
- ✓ Encapsula los datos de nivel superior, a los que añade una cabecera en base a la cual realiza sus funciones.

Para realizar sus funciones, el protocolo IP se sirve de otros protocolos, y los más importantes son los siguientes:

- ✓ ICMP (Internet Control Message Protocol): este protocolo trasmite mensajes entre las entidades de capa de red, como pueden ser errores o situaciones especiales.
- ✓ Protocolos de enrutamiento: estos protocolos le proporcionan al protocolo IP la información necesaria para que decida el camino adecuado para los distintos paquetes. RIP (Routing Internet Protocol) y OSPF (Open Shortest Path First) son dos ejemplos.

Datagrama IP

El protocolo IP realiza sus funciones basándose en la cabecera que añade a los datos de nivel superior. El formato del datagrama[1] IP es el siguiente:

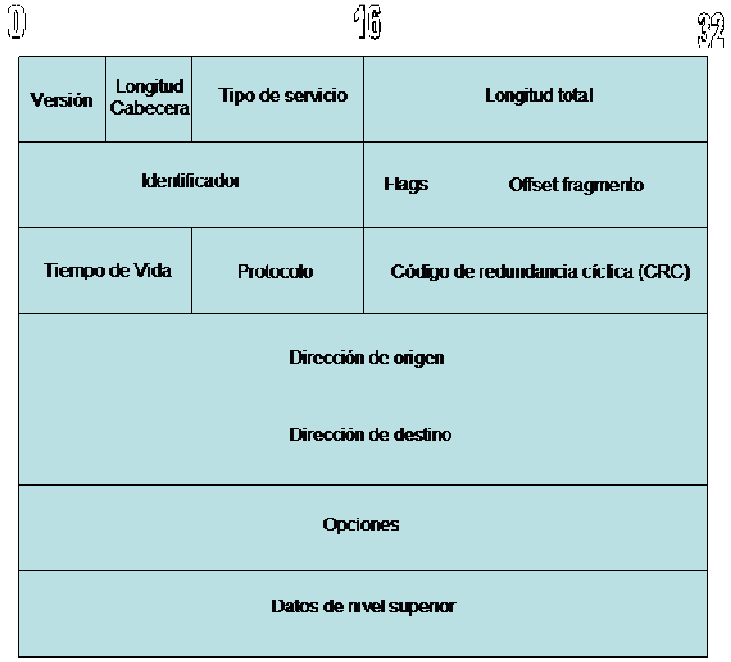

Figura 5. Datagrama IP

Los campos del datagrama son los siguientes:

- ✓ Versión (vers): indica la versión del protocolo IP que se ha usado para formar el paquete.
- ✓ Longitud de la cabecera (hlen): indica la longitud de la cabecera, ya que puede variar dependiendo del campo de opciones.
- ✓ Tipo de servicio: permite que el host especifique que clase de servicio quiere, pudiéndose combinar confiabilidad y velocidad. Para la voz digitalizada es mas importante realizar la entrega de forma rápida que precisa, mientras que para la transferencia de ficheros no importa a que velocidad se realiza la transferencia pero si que esté libre de errores. De los 8 bits, 3 son para el campo de precedencia que en realidad es una prioridad de 0 (normal) a 7 (para los paquetes de control de red). A continuación aparecen los bits de seguridad (alta o baja), retardo (alto o bajo cuando se

[1] Datagrama es como se denominan los paquetes IP.

intenta minimizar el retardo) y rendimiento (normal o alto cuando se intenta maximizar el rendimiento durante la transmisión del datagrama).

- ✓ Longitud total: indica la longitud total del datagrama, expresado en octetos. Este campo tiene 16 bits, por lo tanto el datagrama puede tener una longitud máxima de 65536 octetos (o bytes).
- ✓ Identificador (ID): es un número de secuencia que junto a la dirección origen, la dirección destino y el protocolo de usuario, sirven para que la máquina destino determine a que datagrama pertenece el fragmento que ha recibido. Todos los fragmentos de un datagrama contienen el mismo valor en el campo identificador y este número debe ser único para la dirección origen, la dirección destino y el protocolo de usuario durante el tiempo en el que el datagrama permanece en el conjunto de redes.
- ✓ Flags (3 bits): el primer bit no se utiliza actualmente. El indicador de mas fragmentos (**MF**) cuando vale 1 indica que este datagrama tiene mas fragmentos y toma el valor 0 en el último fragmento. El indicador de no fragmentar (**DF**) prohibe la fragmentación cuando vale 1. Es una orden que se le da a los encaminadores de que no fragmenten el datagrama cuando el destino es incapaz de reensamblarlo. Si este bit vale 1, el datagrama se descartará si se excede el tamaño máximo en una subred de la ruta. Por lo tanto, cuando este bit vale 1, es aconsejable usar encaminamiento por la fuente para evitar subredes cuyo tamaño máximo de paquete sea menor que el tamaño del datagrama.
- ✓ Offset fragmento (13 bits): el datagrama IP puede ser que atraviese redes físicas distintas (con distintas tecnologías y protocolos) desde su origen a su destino. En cada una de estas redes se define un tamaño máximo de trama (MTU) que pueden manejar. Para atravesar alguna de estas redes, si el datagrama es mayor que el MTU de la red que ha de atravesar, el datagrama se ha de fragmentar. Este campo se usa para reconstruir el datagrama en el destino, ya que indica el número de fragmento que ocupa el datagrama que se acaba de recibir en el datagrama original.
- ✓ Tiempo de vida (TTL): Es un contador que sirve para limitar la vida de un paquete. Aunque lo lógico sería pensar que cuenta el tiempo en segundos, en realidad lo que cuenta es el número de saltos de dispositivo de encaminamiento que realiza. Cuando el contador llega a cero, el paquete se descarta y se envía de un paquete al computador origen avisándole. Con este mecanismo se consigue que los datagramas no permanezcan indefinidamente en la red si, por ejemplo, se dañan las tablas de encaminamiento.
- ✓ Protocolo (Protocol): indica el protocolo de capa superior al que pertenecen los datos.
- ✓ Código de redundancia de la cabecera (CRC): este campo contiene un código que sirve para comprobar si hay errores en la cabecera del datagrama. Este valor es verificado y recalculado en cada uno de los dispositivos de encaminamiento.
- ✓ Dirección de origen: es la dirección IP de la estación de origen del datagrama.
- ✓ Dirección de destino: es la dirección IP de la estación de destino del datagrama.
- ✓ Opciones: Contiene las opciones solicitadas por el usuario que envía los datos y se diseñó para que las versiones posteriores del protocolo pudieran

incluir información no considerada originalmente, para que los investigadores pudieran probar cosas nuevas y para que aquellas aquella información que es utilizada pocas veces no tuviera asignada unos bits determinados en la cabecera. Cada una de las opciones empieza en 1 byte que identifica la opción. Algunas de las opciones vienes seguidas de un campo de 1 byte para indicar la longitud de la opción y a continuación uno o mas bytes de datos. Hay seis opciones (Seguridad, Encaminamiento estricto desde el origen, Encaminamiento libre desde el origen,Registrar la ruta, Identificación de secuencia, Marca de tiempo) definidas actualmente.

- ✓ Datos: contiene los datos del nivel superior.

Además hay un campo de relleno, que se introduce si es necesario, para que la longitud de la cabecera sea múltiplo de 32 bits.

Direccionamiento IP

Cada estación conectada a una red TCP/IP se identifica con una dirección única que se utilizará en los campos de direccionamiento de la cabecera. Esta dirección se llama dirección IP. El tamaño de la dirección IP es de 32 bits en la versión 4 del protocolo IP[2].

Cada dirección IP se divide en dos partes: un identificador de subred y un identificador de host. El identificador de subred identifica a la subred a la que pertenece el host, de todas las subredes existentes en la red. El identificador de host identifica a una estación en concreto dentro de la subred que indica el identificador de subred. Para saber cuantos bits de la dirección IP se han destinado al identificador de subred y cuantos al identificador de host se usa la máscara de subred. La porción de la máscara que tiene sus bits a 1 indica que esos mismos bits de la dirección son el identificador de subred, y el resto son el identificador de host. En la siguiente figura se puede ver un ejemplo:

[2] La versión 4 del protocolo IP es la más ampliamente difundida. Ya está funcionando la versión 6, que amplía el rango de direcciones y añade servicio avanzados al protocolo IP.

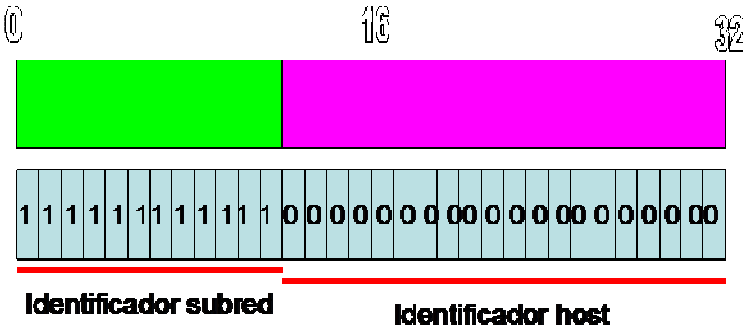

Figura 6. Direccionamiento IP. Dirección y máscara

Hay definidas varias clases de direcciones estándares, que son las que se observan en la siguiente figura:

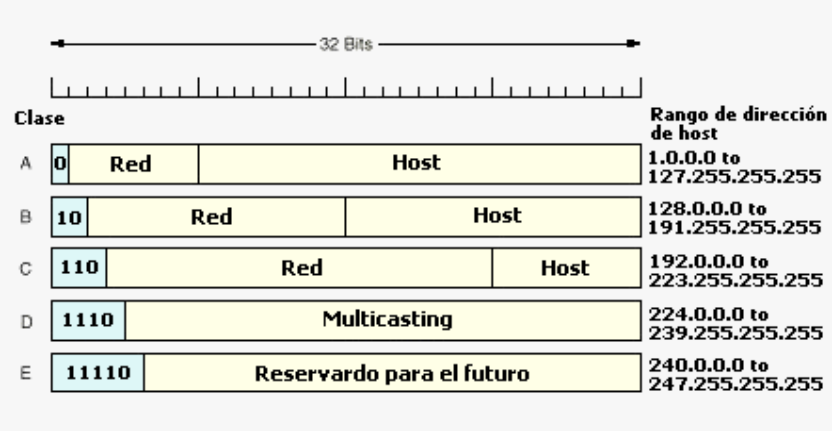

Figura 7. Clases de direcciones IP

A pesar de estas clases de direcciones, uno de estos rangos (por ejemplo, la clase B contiene 65534 direcciones IP válidas, ya que la dirección 0 y la 255 se reservan para la red y para el broadcast respectivamente) se puede dividir en subrangos para hacer un uso más eficiente de la red. Este objetivo se puede alcanzar con distintas máscaras de subred, dependiendo del número de bits que usemos para el identificar de red y para el identificador de host. A esta subdivisión de clases se le conoce como subnetting.

Encaminamiento IP

Cuando un paquete llega a un dispositivo de encaminamiento se debe determinar cual es la dirección del siguiente dispositivo de encaminamiento teniendo en cuenta la dirección IP destino que hay almacenada en el campo correspondiente del paquete y de

la información que hay almacenada en las tablas de encaminamiento. A través de la máscara de subred se averigua la red de destino, y con la información de las tablas de encaminamiento se decide por que interfaz de red hay que enviar el paquete.

Hay que tener en cuenta que es necesario realizar una conversión entre la dirección IP y la dirección MAC (cuando el enlace entre los dos dispositivos de encaminamiento sea una LAN) que se efectúa de manera automática mediante el protocolo ARP.

La tabla de encaminamiento puede ser estática o dinámica. En el primer caso puede contener rutas alternativas que serán utilizadas cuando algún dispositivo de encaminamiento no esté disponible. Las tablas dinámicas son más flexibles cuando aparecen errores o congestión en la red. Estas tablas también pueden proporcionar servicios de seguridad y de prioridad, por ejemplo, para asegurarse que a ciertos datos no se les permita pasar por determinadas redes. Las tablas dinámicas son completadas por los protocolos de encaminamiento, como RIP o OSPF, en base a información que se intercambian los dispositivos de encaminamiento y a algoritmos de cálculo de la ruta más adecuada.

Otra técnica de encaminamiento es el encaminamiento en la fuente. En este caso, como ya comentamos anteriormente, el ordenador origen incluye en la cabecera del paquete la dirección de los dispositivos de encaminamiento que debe utilizar el paquete.

Protocolo ICMP

El protocolo ICMP (Internet Control Messaging Protocol) es usado por el protocolo IP para enviar mensajes entre los nodos de red y los orígenes de los datagramas para notificar situaciones especiales o importantes. Estas situaciones pueden ser, por ejemplo, que el datagrama no puede alcanzar el nodo de destino, que un datagrama se ha de descartar (porque el nodo no tiene suficiente capacidad de almacenamiento) o que el nodo de red puede indicar al origen de los datagramas que puede enrutar los datagramas por una ruta determinada más corta. Otro tipos de mensaje del este protocolo indican el estado de la red.

El tipo de paquete ICMP tiene los siguientes campos:

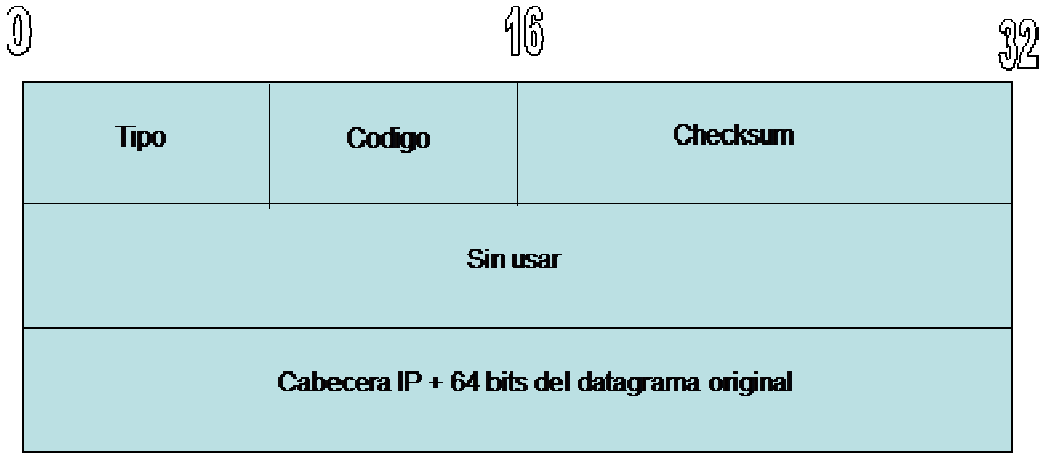

Figura 8. Datagrama ICMP

Los campos son los siguientes:

- ✓ Tipo: indica el tipo de mensaje ICMP que contiene el datagrama. Existen los siguientes:
  - o Echo Reply: 0. Es la respuesta al mensaje de Echo Request.
  - o Source Quench: 4. Mensaje que se envía al origen del datagrama cuando un nodo de red no tiene capacidad de almacenamiento suficiente como para guardar el paquete hasta que sea enrutado hacia el siguiente nodo de red.
  - o Redirect: 5. Este mensaje se envía al origen por parte de un nodo de red para indicarle que puede enviar los datagramas con el mismo destino directamente a otro nodo de red que enrutará estos paquetes (por ejemplo, porque origen y el nuevo nodo están en la misma red) más eficientemente, eliminando un salto en la longitud de la ruta.
  - o Echo Request: 8. Este mensaje pide a un nodo de red que envie una respuesta al origen del mensaje.
  - o Destination Unreachable: 3. La red de destino, según las tablas de enrutamiento, no es alcanzable (número de saltos infinito, etc…).
  - o Time Exceed: 11. El datagrama se descarta porque el tiempo de vida (campo TTL de la cabecera IP) ha llegado a cero. También se emite si el datagrama no se puede reconstruir después de una fragmentación.
  - o Parameter Problem: 12. Mensaje que se envía al origen del datagrama cuando algún nodo de red encuentra algún problema con los campos de la cabecera IP.
  - o Timestamp: 13. Este mensaje envía una marca de tiempo al destino, y requiere una respuesta por parte del destino. Esta marca de tiempo es el momento que se modificó el mensaje por última vez antes de enviarlo.
  - o Timestamp Reply: 14. Respuesta al mensaje de Timestamp, donde se envía la marca de tiempo original más dos marcas de tiempo adicionales. El Received Timestamp indica el tiempo en que se recepcionó el mensaje y el Transmitted Timestamp indica el momento que se envió el mensaje de respuesta.

- Information Request: 15. Este mensaje indica en la dirección de origen la dirección de red y la dirección de destino se deja a cero (por lo que el destino es toda la subred IP).
- Information Reply: 16. Es la respuesta al mensaje de Information Request. La respuesta especifica totalmente las direcciones del mensaje de Information Request.

Con todos estos mensajes se pueden notificar y averiguar condiciones importantes que pueden suceder a lo largo de la red. Por ejemplo, combinando los mensajes Echo Request y Echo Reply se puede averiguar si un nodo de la red en concreto está funcionando o no. También, combinando los mensajes Timestamp y Timestamp Reply se puede averiguar si hay algún problema de recursos en el destino (si la diferencia entre el Received Timestamp y el Transmitted Timestamp es muy grande).

3.1.3. Capa de transporte

La capa de transporte es la primera capa extremo a extremo, es decir, los paquetes del nivel de transporte solo serán recibidos por el destino.

Las funciones principales del nivel de transporte son:

- ✓ Proveer de un servicio de transmisión de datos fiable: control de flujo y errores extremo a extremo.
- ✓ Proveer calidad de servicio a la transmisión de datos a los niveles superiores.
- ✓ Ofrecer una interfaz homogénea a los niveles superiores.
- ✓ Optimizar el uso de la red.

Básicamente hay dos protocolos de nivel de transporte, ya sea la transmisión orientada a conexión o no orientada a conexión. Para el primer caso, el protocolo es TCP (Transmission Control Protocol), y para el segundo caso el protocolo es UDP (User Datagram Protocol).

El direcciónamiento a este nivel se basa en el puerto, que no es más que un número que identifica a la comunicación dentro del mismo equipo, que en general tendrá asignada una sola dirección a nivel de red (dirección IP).

Protocolo UDP

El protocolo UDP provee un servicio de transmisión de datos sin conexión. Se envían los segmentos de datos sin identificación y sin relación unos con otros. Es responsabilidad del nivel superior reconstruir los mensajes si se han de dividir en varios segmentos UDP.

Un ejemplo de uso de UDP es el servicio de DNS (Domain Name Service) que traduce nombres de dominio a direcciones IP.

El formato del paquete UDP se puede observar en la siguiente figura:

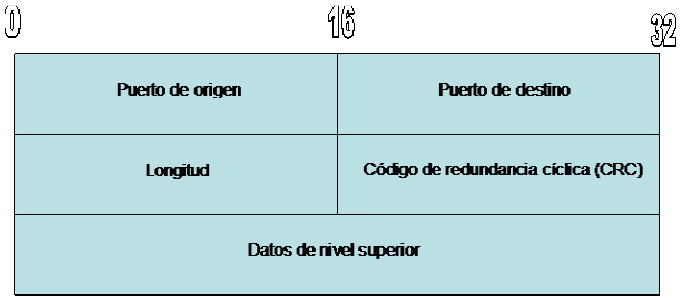

Figura 9. Paquete UDP

Los campos del paquete son los siguientes:

- ✓ Puerto de origen (source port): es el identificativo del puerto de origen del paquete.
- ✓ Puerto de destino (destination port): es el identificativo del puerto de destino del paquete.
- ✓ Longitud (Length): es la longitud del paquete, incluida la cabecera.
- ✓ Código de redundancia cíclica (CRC): es un código de comprobación de errores de todo el paquete, incluidos los datos.

Protocolo TCP

El protocolo TCP proporciona a los niveles superiores de la red una transmisión de datos fiable, sin errores y orientada a la conexión. Por lo tanto, antes de comenzar la transmisión de datos hay que establecer una conexión entre los equipos de origen y destino.

El nivel superior (generalmente el nivel de aplicación) entrega los mensajes a enviar a la capa de transporte, TCP los fragmenta en partes de no más de 64K octetos (segmentos, una vez que se les añade la cabecera TCP) y los envía por la conexión que ha establecido con el destino. En el destino se asegurará la integridad de estas partes, se reordenarán adecuadamente (ya que IP no garantiza la entrega en orden de los datagramas) y se pasará el mensaje al nivel superior. TCP también provee de métodos de recuperación de errores y de segmentos perdidos.

La cabecera que TCP utiliza para implementar todos estos mecanismos es la siguiente:

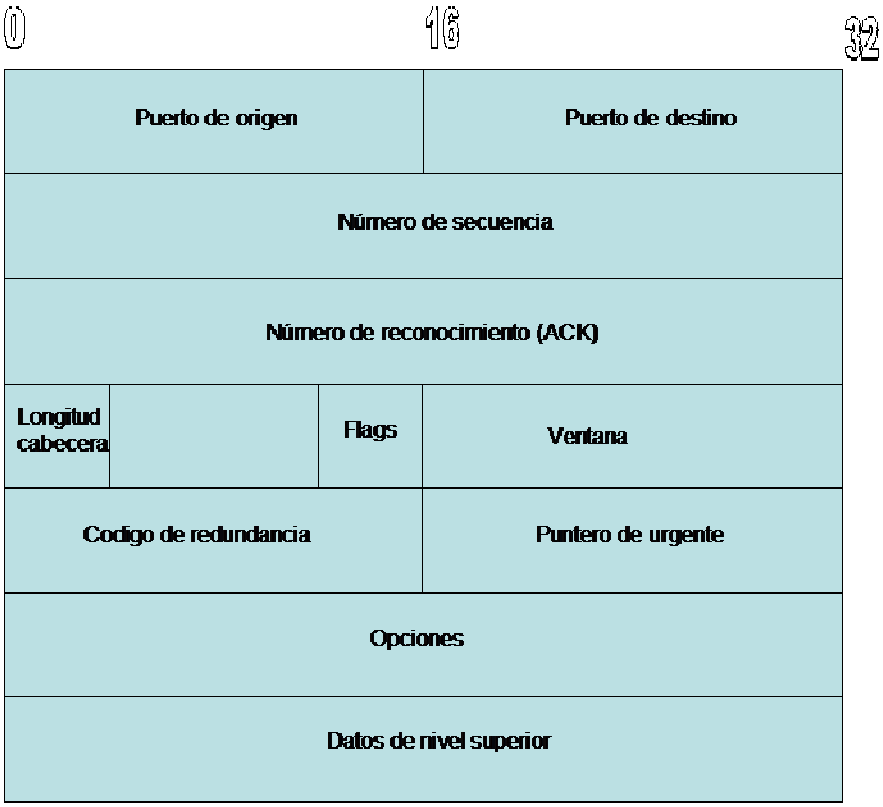

Figura 10. Cabecera TCP

Los campos que aparecen en la figura anterior son los siguientes:

- ✓ Puerto de origen: identificador de la conexión a la que pertenece el segmento TCP en el equipo de origen.
- ✓ Puerto de destino: identificador de la conexión a la que pertenece el segmento TCP en el equipo de destino.
- ✓ Número de secuencia: identifica la posición del primer octeto del campo de datos dentro del mensaje del nivel superior. TCP numera octetos, no los segmentos.
- ✓ Número de reconocimiento (ACK): contiene el número del octeto que espera recibir el destino, por lo tanto, indica al origen de los datos que el octeto ACK-1 ha llegado correctamente al destino.
- ✓ Longitud de la cabecera: indica la longitud de la cabecera del segmento TCP en palabras de 32 bits.
- ✓ Flags: son seis bits que indican diferentes características del segmento TCP, Se utilizan, por ejemplo, en la identificación los paquetes de establecimiento de conexión. Son los siguientes:
  - o **urg:** Hay datos urgentes y en el campo "puntero urgente" se indica el número de datos urgentes que hay en el segmento.
  - o **ack:** Indica que tiene significado el número que hay almacenado en el campo "número de reconocimiento".
  - o **psh:** Sirve para invocar la función de carga (push). Como se ha comentado anteriormente con esta función se indica al receptor que debe pasar a la aplicación todos los datos que tenga en la memoria intermedia sin esperar a que sean completados. De esta manera se

consigue que los datos no esperen en la memoria receptora hasta completar un segmento de dimensión máxima. No se debe confundir con el indicador URG que sirve para señalar que la aplicación ha determinado una parte del segmento como urgente.
    - **rst:** Sirve para hacer un reset de la conexión.
    - **syn:** Sirve para sincronizar los números de secuencia.
    - **fin:** Sirve para indicar que el emisor no tiene mas datos para enviar.
- ✓ Ventana: Indica cuantos bytes tiene la ventana de transmisión del protocolo de control de flujo utilizando el mecanismo de ventanas deslizantes. A diferencia de lo que ocurre en los protocolos del nivel de enlace, en los que la ventana era constante y contaba tramas, en el TCP la ventana es variable y cuenta bytes. Contiene el número de bytes de datos comenzando con el que se indica en el campo de confirmación y que el que envía está dispuesto a aceptar.
- ✓ Código de redundancia: campo de control que se utiliza para detectar los posibles errores que se hayan producido en la transmisión y manipulado del segmento.
- ✓ Puntero de urgente: Cuando el indicador URG está activo, este campo indica cual es el último byte de datos que es urgente. De esta manera el receptor puede saber cuantos datos urgentes llegan. Este campo es utilizado por algunas aplicaciones como telnet, rlogin y ftp.
- ✓ Opciones: Si está presente permite añadir una única opción de entre las siguientes:
    - Timestamp: para marcar en que momento se transmitió el segmento y de esta manera monitorizar los retardos que experimentan los segmentos desde el origen hasta el destino.
    - Aumentar el tamaño de la ventana.
    - Indicar el tamaño máximo del segmento que el origen puede enviar.

Como hemos comentado anteriormente, TCP es un protocolo orientado a la conexión, es decir, que para transmitir datos desde un origen a un destino ha de establecer una conexión entre origen y destino. El proceso de apertura de una conexión es el siguiente:

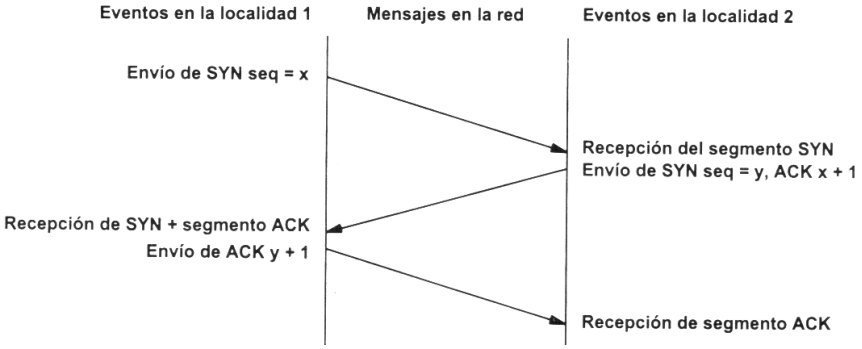

Figura 11. Establecimiento de una conexión con TCP

Con ello conseguimos, entre otras cosas, reservar recursos para gestionar esta conexión y asignar los números de puerto tanto en origen como en destino, que identificarán a los segmentos que pertenecen a esta conexión, de todos los segmentos que el nivel de red pase al nivel de transporte.

Obviamente, al final de la comunicación se realiza un cierre de conexión, cuyo flujo de paquetes se puede ver en la siguiente figura:

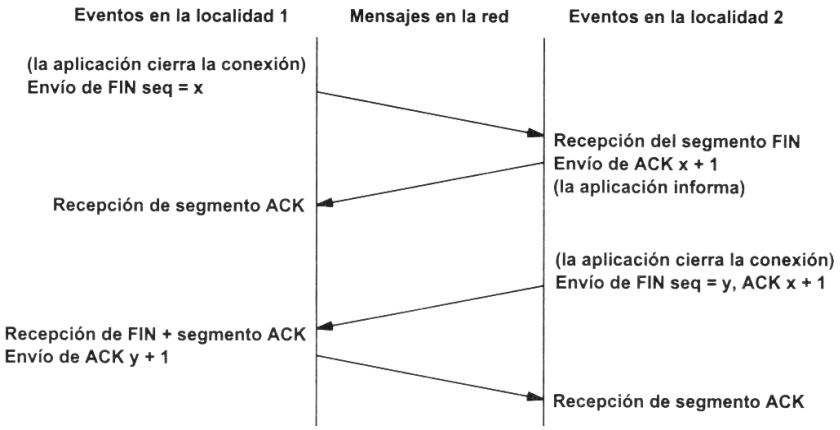

Figura 12. Cierre de una conexión con TCP.

Como observamos con estas figuras, tanto en el establecimiento como en el cierre de una conexión juegan un papel muy importante los flags de la cabecera, así como el número de reconocimiento (ACK).

3.1.4. Resumen

Hemos visto una breve descripción de los protocolos principales que integran la arquitectura TCP/IP, y que nuestro capturador de red va a identificar.

Por un lado tenemos el protocolo IP y el protocolo ICMP, ambos de la capa de red, que hacen que los paquetes de datos lleguen del origen al destino a través de múltiples redes, que pueden ser de tecnologías distintas. Ambos protocolos proporcionan una gran robustez a la arquitectura, ya que son muy adaptables y resistentes a fallos.

Por otro lado tenemos dos protocolos de la capa de transportes, que es la primera capa extremo a extremo de la arquitectura. UDP proporciona una transmisión sin conexión, sencilla, de paquetes individuales, y por tanto muy rápida. TCP proporciona una comunicación fiable, libre de errores a través de la red y orientada a la conexión. Sobre ambos protocolos podemos desarrollar aplicaciones, y con un capturador de

paquetes como el descrito en esta memoria se pueden averiguar cosas interesantes del funcionamiento de los protocolos descritos.

Capitulo 4. Diseño del capturador de paquetes de red

4.1. Requerimientos del capturador

Vamos a estructurar los requerimientos del programa a diseñar en dos partes: primero describiremos los requerimientos de funcionamiento que exigiremos al programa, y segundo, los requerimientos técnicos que vamos a necesitar en las máquinas donde queramos ejecutar el programa. Hemos evaluado los siguientes capturadores disponibles en el mercado:

- ✓ Ethereal (www.ethereal.com/): capturador gratuito de código abierto, muy completo, disponible para Windows y para Linux/Unix.
- ✓ EtherSnoop(www.arechisoft.com/): capturador que tiene una versión gratuita y otra de pago. Disponible solo para Windows.
- ✓ Colasoft Capsa (www.colasoft.com/products/capsa/): capturador de pago muy completo para Windows.

Requerimientos de funcionamiento

Después del repaso que se han dado a los capturadores actuales en el mercado, los requerimientos de negocio del capturador son los siguientes:

- ✓ Debe ser posible de elegir el interfaz de red de donde capturar los paquetes.
- ✓ Los paquetes capturados deben guardarse en un fichero, para luego hacer análisis más exhaustivos sobre los datos.
- ✓ Los datos en el fichero han de guardarse en texto plano.
- ✓ La captura de paquetes ha de ser on-line, y que pueda ser consultada mientras se capturan paquetes.
- ✓ El uso del programa ha de ser muy sencillo.
- ✓ Se han de capturar los paquetes de los protocolos IP, TCP, UDP e ICMP.

Con estos requerimientos tenemos un capturador de paquetes que proporciona datos muy valiosos sobre las cabeceras de cada paquete.

Requerimientos técnicos de uso

En cuanto a requerimientos técnicos, son los siguientes:

- ✓ El programa está compilado para plataformas Windows XP en adelante.

- ✓ Se necesita tener instalado las librerías Winpcap (ver [2]).

4.2. Diseño del capturador

Para cumplir los requisitos marcados en el apartado anterior, hemos recurrido a la librería WinPcap. Con esta librería tenemos acceso a los interfaces de red y a los paquetes capturados, a partir de los cuales podemos desarrollar las funcionalidades del capturador. Por lo tanto se ha de tener instalada la librería WinPcap (ver [2]), que se puede descargar gratuitamente.

El diagrama de flujo que sigue el programa diseñado se puede ver en la siguiente figura:

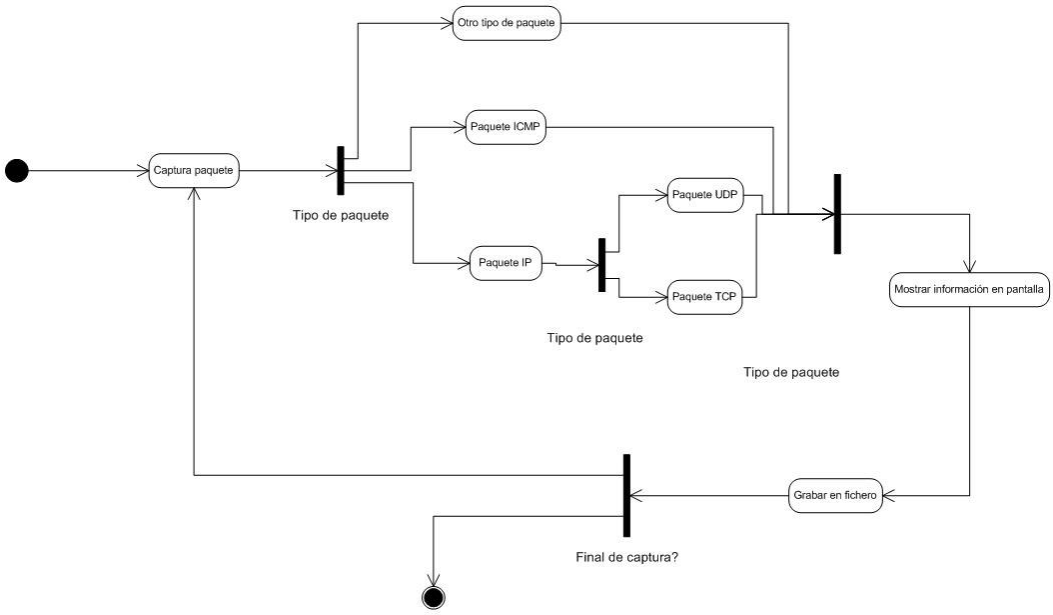

Figura 13. Diagrama de flujo del capturador de paquetes

Como vemos, después de la captura de un paquete se determina su tipo. Dependiendo del tipo de paquete se muestra por pantalla (y se graba en un fichero) una información determinada, extraída de la cabecera del paquete.

Otras decisiones de diseño son las siguientes:

- ✓ Se ha decidido concentrar todo el código en un solo fichero. Solo se implementa la función de tratamiento de los paquetes capturados, a parte del programa principal, que inicializa las variables y pide los datos necesarios.

- ✓ Se ha reutilizado código de [2]. Básicamente ha sido la estructura de captura y las funciones a las que había que llamar.
- ✓ Los comentarios se han realizado en inglés para facilitar el intercambio del código.
- ✓ Se ha creado un fichero de cabecera con todas las definiciones de tipos utilizados, para facilitar su uso posterior.
- ✓ La tarjeta de red se coloca en modo promiscuo (es decir, que captura todos los paquetes que pasan por su segmento de red) siempre en el programa, y no se da opción al usuario a cambiar esta propiedad.
- ✓ Todas las capturas se vuelcan en un fichero, que ha de estar en el mismo directorio que el ejecutable del programa de captura.

4.3. Resumen

Con este diseño se ha realizado un programa que cumple los requerimientos establecidos, pero además es escalable y fácilmente ampliable. También pueden usarse sus definiciones de datos desde otros programas, simplemente incluyendo su fichero de cabecera. Los comentarios en inglés facilitarán su difusión.

Los ficheros implicados son:

- ✓ Main.cpp: es el fichero principal que contiene todo el código programado.
- ✓ Capturador.h: es el fichero de cabecera donde están todas las definiciones de los tipos de cabecera que reconoce el programa de captura.

Capitulo 5. Uso del capturador de paquetes de red

Este capturador de paquetes se ha diseñado para su ejecución en plataformas Windows XP en adelante. Por lo tanto, la máquina donde se ejecute debe de ejecutar este sistema operativo, además de tener instalada la librería WinPcap (ver [2] y capítulo 4).

5.1. Funcionamiento del capturador

El funcionamiento del capturador es muy sencillo. El programa ejecutable se llama capturador.exe, y se puede invocar de dos formas:

- ✓ Sin parámetros: el volcado de datos se hará en un fichero llamado dumfile.txt.
- ✓ Con 1 parámetro: se ha de indicar el nombre del fichero donde se quiere que se haga el volcado de los datos de la captura. Este fichero ha de estar situado en el mismo directorio que el programa ejecutable. Si no existe, el programa lo crea. El volcado de datos se hace a continuación de los datos que pueda ya haber en el fichero (no se borran datos previos, muy útil si queremos guardar en el mismo fichero varias capturas para luego compararlas).

En ambos casos, los paquetes capturados también se muestran por pantalla.

Una vez invocado el programa aparecerá la siguiente pantalla:

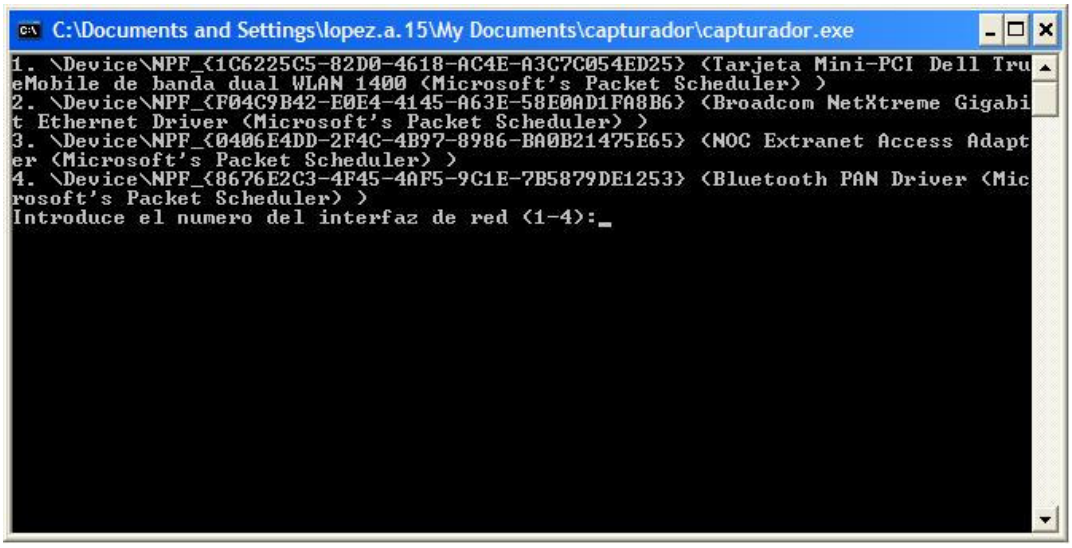

Figura 14. Primera pantalla del programa

En esta primera pantalla aparece un listado de los interfaces de red disponibles en nuestro ordenador. Debemos seleccionar aquel sobre el que queremos hacer la captura.

Una vez seleccionado el interfaz de red, el programa nos pregunta que tipo de paquetes queremos ver, tal y como aparece en la siguiente pantalla:

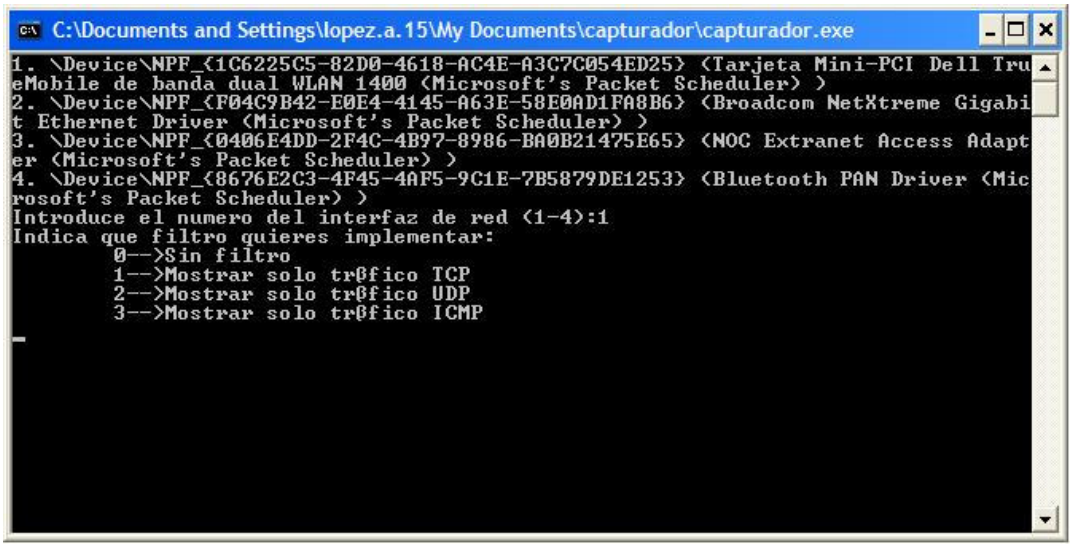

Figura 15. Tipos de filtro de paquetes que incluye el programa

Seleccionamos el que queremos y presionamos Enter, y el programa comenzará la captura de paquetes, tal y como se observa en la siguiente figura:

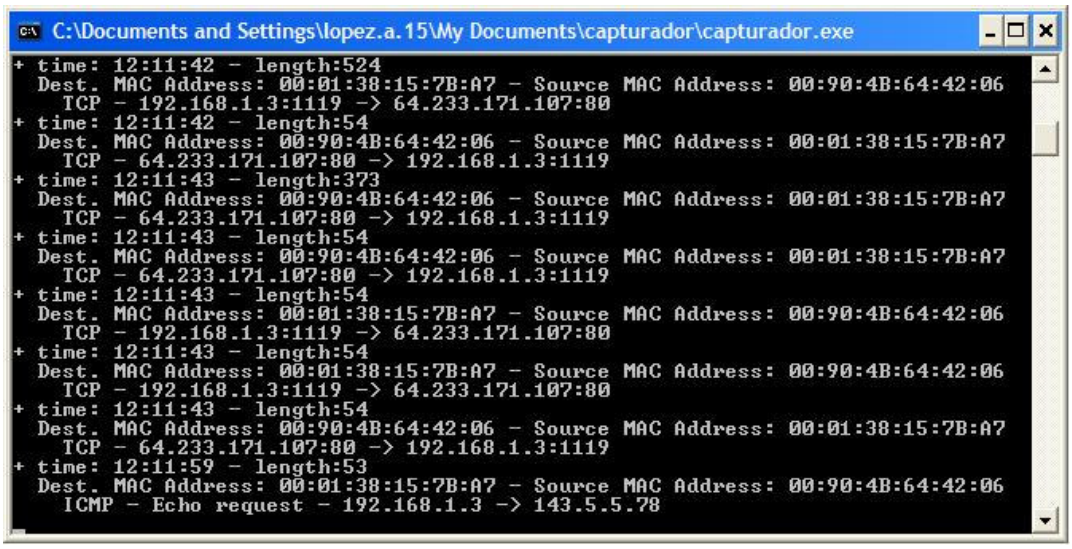

Figura 16. Ejemplo de captura de paquetes

Esta misma salida se guardará en el fichero que hayamos especificado al pasarlo como parámetro al programa, o en dumpfile.txt si hemos dejado el fichero por defecto.

Para salir del programa, simplemente interrumpir la ejecución con Ctrl+C.

5.2. Resumen

Como hemos visto, la utilización del programa es muy sencilla. Éste era uno de los objetivos que queríamos alcanzar con este capturador, frente a la complejidad de otros capturadores. También, en el fichero de volcado tenemos los paquetes capturados que se observan on-line en pantalla, para su futuro análisis, cumpliendo así el resto de los objetivos.

Capítulo 6. Conclusiones

6.1. Conclusiones de la realización

El objetivo de este trabajo final de carrera era el diseño y la implementación de un capturador de paquetes de red. Este capturador debía cumplir los siguientes requisitos:

- ✓ Debe ser posible de elegir el interfaz de red de donde capturar los paquetes.
- ✓ Los paquetes capturados deben guardarse en un fichero, para luego hacer análisis más exhaustivos sobre los datos.
- ✓ Los datos en el fichero han de guardarse en texto plano.
- ✓ La captura de paquetes ha de ser on-line, y que pueda ser consultada mientras se capturan paquetes.
- ✓ El uso del programa ha de ser muy sencillo.
- ✓ Se han de capturar los paquetes de los protocolos IP, TCP, UDP e ICMP.

Como hemos observado a lo largo de esta memoria, estos requisitos se han cumplido perfectamente. Se ha realizado un análisis de las cabeceras de los protocolos requeridos, siguiendo lo especificado en el capítulo 3.

Por lo tanto, tenemos un programa capturador de paquetes que nos puede servir para ver el tráfico que tenemos en nuestra red, y detectar con el análisis de este tráfico si existe algún problema en la red y seguro que nos dará pistas para poder hallar una solución.

6.2. Posibilidades de ampliación

Como punto final de esta memoria, vamos a citar varias líneas de ampliación del capturador de paquetes aquí descrito. Como hemos comentado anteriormente, el diseño se ha realizado con el objetivo de que el programa sea escalable y fácilmente ampliable. Las posibles ampliaciones a hacer son las siguientes:

- ✓ Dotar al programa de un entorno gráfico para Windows. Este entorno gráfico dará facilidad de uso cuando crezcan las posibilidades del capturador.
- ✓ Ampliar el número de protocolos que reconoce el capturador. Actualmente el capturador de paquetes reconoce los protocolos más importantes, pero se puede ampliar para que reconozca muchos más.

- ✓ Crear un módulo que genere estadísticas a partir de los paquetes capturados. Por ejemplo, porcentaje de paquetes de un determinado protocolo, o porcentaje de paquetes mal formados, etc…
- ✓ Crear un entorno de análisis off-line de los ficheros de captura, para reproducir situaciones anteriores, o repetir análisis sobre los mismos datos.
- ✓ Analizar los datos de aplicación, para determinar de que tipo de aplicación son y que datos llevan, así como funcionalidades específicas de algún protocolo (por ejemplo, detectar un establecimiento de conexión TCP).

Capítulo 7. Bibliografía

[1] *Redes de Ordenadores*, Andrew S. Tanenbaum. Ed. Prentice Hall. ISBN: 0-13-162959-X

[2] Librería WinPcap, http://winpcap.polito.it/

[3] Modelo de referencia OSI, http://www.geocities.com/txmetsb/el_modelo_de_referencia_osi.htm

[4] Arquitectura TCP/IP, http://eia.udg.es/~atm/tcp-ip/index.html

[5] Arquitectura TCP/IP, http://pegaso.ls.fi.upm.es/~lmengual/anexo/

[6] Arquitectura TCP/IP, tutorial, http://www.uca.edu.sv/investigacion/tutoriales/tcp-ip.html

[7] Relación modelo OSI-TCP/IP, http://www.htmlweb.net/redes/osi/osi_1.html

[8] Protocolo IP, RFC 0791, http://www.cse.ohio-state.edu/cgi-bin/rfc/rfc0791.html#sec-2.1

[9] Protocolo IP, http://eia.udg.es/~atm/tcp-ip/tema_4_5.htm

[10] Protocolo ICMP, RFC 0792, http://www.cse.ohio-state.edu/cgi-bin/rfc/rfc0792.html

[11] Protocolo TCP, RFC 0793, http://www.rfc-es.org/getfile.php?rfc=0793

[12] Protocolo TCP, http://www.saulo.net/pub/tcpip/b.htm#3-3

[13] Protocolo UDP, RFC 768, http://www.faqs.org/rfcs/rfc768.html

[14] Asignación de numeración de protocolos, RFC 0790, http://www.cse.ohio-state.edu/cgi-bin/rfc/rfc0790.html

[15] Tipos de redes Ethernet, http://www.webopedia.com/quick_ref/EthernetDesignations.asp

[16] Historia de Internet, http://galeon.hispavista.com/epymes/enlaces760579.html

[17] Compilador C++, http://www.bloodshed.net/

[18] Tutorial de C++, http://www.cplusplus.com/doc/tutorial/

[19] Ethereal, http://www.ethereal.com/

[20] Visual C++ .NET, Jose Angel Jimenez Vadillo. Ed. Anaya. ISBN: 84-415-1461-5