

DISEÑO DE UNA LIBRERÍA EN LENGUAJE "C" PARA MANEJAR TARJETAS ETHERNET EN LAS PRÁCTICAS DE REDES DE COMPUTADORES

Verónica Medina, Francisco Pérez, Sergio Martín, Jaime Benjumea y Daniel Carretero
Departamento de Tecnología Electrónica
Facultad de Informática y Estadísticas
Universidad de Sevilla
Avda. Reina Mercedes s/n,
41012 - SEVILLA -
Tlf:(95)4557095, Fax:(95)4552764
e_mail:anamed@obelix.cica.es

RESUMEN.- En este trabajo se presenta el diseño de una librería en lenguaje "C" que facilita el desarrollo de las prácticas de laboratorio de la asignatura de 4º curso de Redes de Computadores en la licenciatura de Informática en la Universidad de Sevilla.

1.- INTRODUCCIÓN

En el laboratorio de Redes de Computadores de la Facultad de Informática y Estadística de la Universidad Sevilla, hay instalada una red de ordenadores con PC's que utilizan tarjetas ethernet para la conexión de los diferentes equipos que la integra. Dichas tarjetas son de múltiples fabricantes, por lo que aunque cumplen las características definidas en la norma, sus componentes hardware son distintas, y su interacción con el resto del sistema (en este contexto indica el conjunto de hardware y software que forma parte de un PC) difiere de unas a otras. Además, el fabricante no suele describir cómo se realiza dicha interacción, lo que casi imposibilita elaborar unas prácticas que utilicen las tarjetas ethernet para realizar conexiones multipunto.

También cabe destacar que las prácticas de la asignatura de Redes de Computadores son de laboratorio abierto, o sea, el alumno puede utilizar el laboratorio para realizar las prácticas cuando quiera y durante el tiempo que el estime oportuno. Esto presenta el siguiente problema: Cuando se acerca la fecha de entrega de las prácticas, el laboratorio está congestionado debido a que existe un número reducido de equipos. Este problema se solucionaría si el alumno además de poder llevar a cabo las prácticas en la red instalada en el laboratorio, pudiera hacer las prácticas en otra red que estuviera instalada dentro y fuera de la Facultad, por ejemplo en el Centro de Cálculo, en los seminarios de los departamentos, etc.

2.- FUNCIONAMIENTO DE LA RED ETHERNET

Ethernet es un estándar de red local cuya especificación fue publicada en 1980 por Xerox, Intel y Digital. Esta especificación con algunos cambios en el formato de las tramas constituye el estándar IEEE 802.3. Ethernet se organiza en torno al concepto de niveles propuesto por el modelo de referencia OSI, cubriendo el subnivel de control de acceso al medio o MAC (medium access control) que forma parte del nivel de enlace de datos y el

nivel físico. El subnivel MAC se encarga de la transmisión/recepción de los datos, de la gestión de acceso al medio para la transmisión/recepción, de la codificación/decodificación de los datos, y del acceso al canal. Este subnivel es común para los distintos medios físicos (cable coaxial, fibra óptica, ...). Ethernet utiliza una topología tipo bus, en el que todas las estaciones se consideran iguales; las estaciones compiten por el canal en las mismas condiciones. Debido a que más de una estación puede acceder al canal en el mismo instante, se debe utilizar algún mecanismo de control del bus (de acceso al medio). Además, para identificar a las distintas estaciones que estén en el bus se utiliza un número (de 48/16 bits para 802.3 y 48 bits en Ethernet) que es único para cada estación y que la distingue del resto.

La técnica de acceso al medio que se utiliza para el acceso al bus es la CSMA/CD (Carrier Sense Multiple Access/Colision Detection). En esta técnica cada estación que integra el bus antes de transmitir debe monitorizar el canal para ver si el canal está ocupado. Si el canal está inactivo, cualquier estación que tenga datos para transmitir puede enviarlos. Si está ocupado, las estaciones deben dejar el canal para la estación que lo estaba utilizando. Si dos estaciones detectan el canal libre y transmiten simultáneamente, se produce una colisión que es detectada por las estaciones que colisionan. En este caso, las estaciones deben abortar la transmisión, esperar un tiempo aleatorio y competir de nuevo por el canal. El tiempo de espera aleatorio reduce la posibilidad de que ocurra una nueva colisión, ya que es poco probable que las estaciones que compiten generen el mismo tiempo de espera.

A continuación se describe como la tarjeta ethernet resuelve los problemas de acceso al medio especificados en la norma de una forma transparente a sus usuarios.

2.1.- TARJETA ETHERNET

La tarjeta ethernet [1] trabaja como un intermediario entre el ordenador y la red. Este intermediario está siempre atento a todo lo que pase por la red y procura comportarse como una estación dentro de la norma 802.3 (Figura 1). Por tanto debe estar al tanto de colisiones, lectura de las tramas, etc. A su vez debe atender las peticiones de ordenador, recoger las tramas que el usuario quiere mandar y ponerlas en la red, también debe coger las tramas que tengan como destino el propio ordenador y dárselas al mismo. De esta manera, para usar la red no necesitamos saber las características complicadas de ésta, sino solamente tratar con un intermediario que sabe cómo debe actuar dentro de la norma 802.3 (ethernet).

La tarjeta ethernet se instala en el PC en una dirección base de E/S (entrada/salida) a partir de la cual se accede a sus registros. Esta dirección puede no ser la misma para las tarjetas instaladas en diferentes ordenadores. La dirección base usual es la 280h. La tarjeta está diseñada para permitir un anillo de recepción (buffer de recepción), de tal forma que almacena en ese anillo las tramas que llegan, mientras sigue leyendo las que van viniendo. El anillo se implementa a partir de una dirección de memoria que usualmente es la D000:0000h. Para el manejo de las interrupciones se suele utilizar la IRQ 3 y el vector de interrupciones 0x0B, pero puede haber conflictos si se tienen instalados puertos de comunicación serie. Las configuraciones más usuales son:

<u>BASE</u>	<u>RAM</u>	<u>MEMORIA</u>	<u>IRQ</u>	<u>VECTOR</u>
280h	D000:0000	16k	3	0Bh
300h	CA00:0000	8k	5	0Dh
2A0h	CC00:0000	16k	10	12h
240h	D800:0000	32k	2	0Ah

Las tarjetas [3] más antiguas requieren que las configuraciones se realicen modificando la posición de algunos jumpers o switches. Las más modernas se configuran por software y son capaces de autoconfigurarse, detectando posibles conflictos con direcciones de E/S y vectores de interrupción que ya estén ocupados.

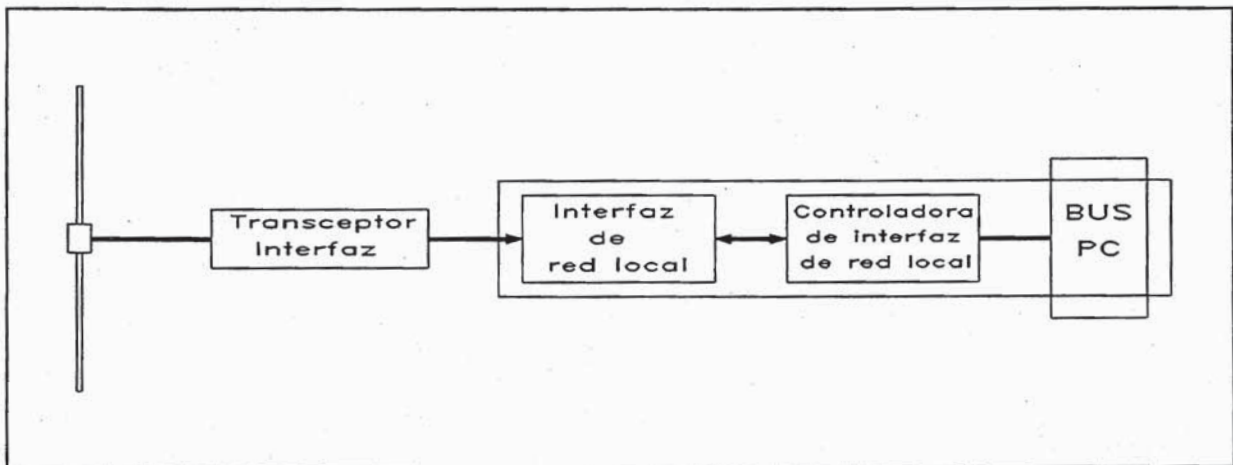


Figura 1.- Tarjeta Ethernet.

Una tarjeta tiene un identificador único que lo diferencia desde el momento de su fabricación del resto de las tarjetas. Dicho identificador se compone de 6 octetos y corresponde a la dirección MAC (ethernet), o dirección física. Los tres primeros corresponden al número del fabricante (p.e. 60:01:3B:56:7C:F1, 60013B es el número de fabricante), el resto es el número de tarjeta dentro del fabricante. De esta forma no existen conflictos entre direcciones de varias tarjetas. Normalmente la tarjeta espera tramas (datos) que tengan como dirección destino su número de identificación, pero existen otros identificadores especiales:

- **Identificador MULTICAST**, se diferencia por empezar con el bit más significativo a 1. El resto de los bits se usan según se haga el multicast. Sirve para realizar grupos dentro de una red.
- **Identificador BROADCAST**, está compuesto por todos su bit a 1. Es útil cuando se quiere que una trama tenga como dirección de destino todas las estaciones de la red.

La estructura de una trama o paquete ethernet es la siguiente:

Preámbulo	Indicador de comienzo	Dirección de destino	Dirección de origen	Tipo	Datos	Relleno	Redundancia
(7)	(1)	(6)	(6)	(2)	(46 -	1486)	(4)

Los primeros ocho bytes se utilizan para la sincronización. La dirección fuente y origen son 12 bytes que identifican a las tarjetas destino y origen del paquete, respectivamente. El tipo (conocido también por Ethertype) se utiliza para indicar el tipo de protocolo que se sitúa por encima del nivel MAC (0x0008h: protocolo ip, 0x0608: protocolo arp, 0x0618: protocolo RC-ARP, ...). Los datos que se envíen tienen que tener una longitud mínima de 46 bytes y máxima de 1486 bytes, el relleno se utiliza para completar datos que ocupen menos de 46 bytes. El tamaño mínimo de un paquete es 60 y el máximo de 1500 bytes. Para la detección de errores utiliza un CRC (código de redundancia cíclica) de 32 bits (4 bytes).

Un usuario de la tarjeta trabajaría sólo con los datos de la trama que corresponden a la Dirección de destino, la Dirección de origen, el Tipo de protocolo, y los Datos, el resto de los campos de la trama los inserta la tarjeta de una forma automática y transparente.

Dependiendo del tipo de tramas que se lean, la tarjeta puede trabajar en diferentes modos de operación:

- *Modo Normal.* Lee los paquetes o tramas que lleven su dirección física, las broadcast y las direcciones multicast que coincidan con la suya.
- *Modo sin Broadcast.* Lee los paquetes que llevan su dirección física, y aquellos cuya dirección multicast coincida con una suya. (No toma paquetes Broadcast).
- *Modo Promiscuo.* Lee todos los paquetes.

Con toda esta información, si se quiere realizar unas rutinas que sirvan para enviar y recibir paquetes ethernet de la tarjeta, se necesita conocer las características hardware de la misma, o sea, como interactúa la tarjeta ethernet con el resto del sistema. Estas características suelen ser desconocidas para el comprador de la tarjeta, por lo que se encuentra una barrera a la hora de diseñar programas que manejen dichas tarjetas. Sin embargo, la empresa *FTP Software, Inc* [2] tiene acceso a esa información y con ella crea unos drivers (denominados *Packet Driver*) que tienen una interfaz común, independiente del modo en que la tarjeta ethernet interactuare con el sistema. En la Figura 2 se observa cómo un driver oculta el funcionamiento de la tarjeta a cualquier programa que utilice dicha interfaz.

3.- INTERFAZ CON LA TARJETA ETHERNET

En el apartado anterior se ha visto que para manejar la tarjeta ethernet se podría optar por dos soluciones:

- 1.- Manejar directamente el hardware.
- 2.- Utilizar un driver que maneje la tarjeta.

La primera solución sería la óptima, si conociéramos el hardware de todas las tarjetas que tiene el laboratorio y este no fuera a cambiar, ya que el software se diseñaría a medida. La segunda solución es la viable, pues se tiene acceso al driver de la tarjeta (las tarjetas más modernas lo incluyen en el disco de configuración), con lo que conociendo la interfaz con el driver se puede manejar la tarjeta. Además la instalación del driver de una tarjeta determinada es muy fácil, ya que solo requiere conocer el puerto de E/S de la tarjeta, la IRQ que usa para el manejo de las interrupciones, la dirección base de la RAM donde se almacenan las tramas que llegan, y la interrupción software que se va a utilizar (p.e. 0x61). Muchos drivers se instalan automáticamente.

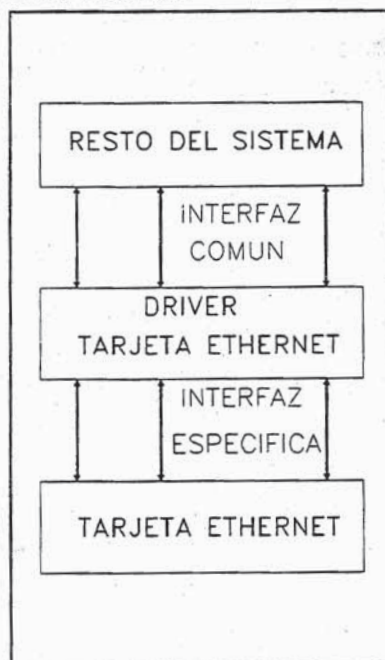


Figura 2.- Interfaz driver resto sistema

Aunque el manejo de la interfaz del driver no es muy difícil, requiere que el alumno tenga conocimiento de algunas características hardware y software (independientes de la tarjeta) que no aportan nada a las prácticas de Redes de Computadores. Por ello se ha diseñado una librería compuesta por una serie de rutinas en lenguaje "C" que sirven de interfaz común entre el programa del alumno y el driver (Figura 3). De esta forma el alumno con sólo

conocer el driver de la tarjeta que vaya a utilizar y enlazando la librería diseñada junto con su programa, podrá realizar las prácticas de redes en el lugar que estime oportuno. En resumen, para poder utilizar cualquier tarjeta ethernet lo primero que hay que hacer es buscar el driver que conozca el hardware de la misma y utilizar la librería para controlar a la tarjeta ethernet.

4.- LIBRERÍA ETHERNET

El nombre librería ethernet se refiere al conjunto de funciones que sirven para manejar la tarjeta ethernet a nivel de usuario, sin tener en cuenta ni las características de la interfaz de los drivers, ni el hardware de las tarjetas. El usuario que utilice la librería sólo tiene que preocuparse de buscar la función que le dé el servicio deseado, de tal manera que con el nombre de la función y los parámetros de la misma obtenga la funcionalidad deseada.

Como se ha mencionado, la librería se ha diseñado utilizando el lenguaje de programación "C". Esta consta las siguientes funciones:

- *instalar_protocolo*. Esta función sirve para habilitar al driver para la recepción de paquetes asociados al protocolo especificado como parámetro. Si el protocolo que se indica como parámetro es cero se inicia la recepción de tramas para todos los protocolos.
- *desinstalar_protocolo*. Desinstala un protocolo instalado por la función *instalar_protocolo*. El protocolo se pasa como parámetro.
- *desinstalar_driver*. Libera todos los protocolos instalados.
- *obtener_estadisticas*. Esta función sirve para obtener las estadísticas sobre el funcionamiento de la tarjeta ethernet.
- *obtener_direccion*. Sirve para conocer la dirección física de la tarjeta local, o sea, la dirección ethernet.
- *establecer_modos_recepcion*. Como se ha mencionado antes, existen diferentes modos de direccionamiento que valen para saber las tramas ethernet que se deben recibir, los modos que se pueden configurar mediante esta función son los siguientes:
 - 1.- Ningún paquete. Se deshabilita la recepción de paquetes.
 - 2.- Sólo aquellas tramas dirigidas a la dirección de la tarjeta local.
 - 3.- Igual que la 2 más los broadcast.
 - 4.- Igual que la 3 más los multicast establecidos con la función *establecer_multicast* que se verá a continuación.
 - 4.- Igual a 3 más los multicast.
 - 5.- Todos los paquetes.
- *obtener_modos_recepcion*. Devuelve un código que indica los paquetes que se pueden recibir (el modo en que la tarjeta está configurada).
- *establecer_multicast*. Establece como lista de direcciones multicast la que se le pase como parámetro. Esto es útil para poder dividir la red en trozos y que sólo se puedan recibir tramas de las direcciones que uno especifique en la lista multicast.
- *obtener_multicast*. Devuelve la lista multicast que se está utilizando.

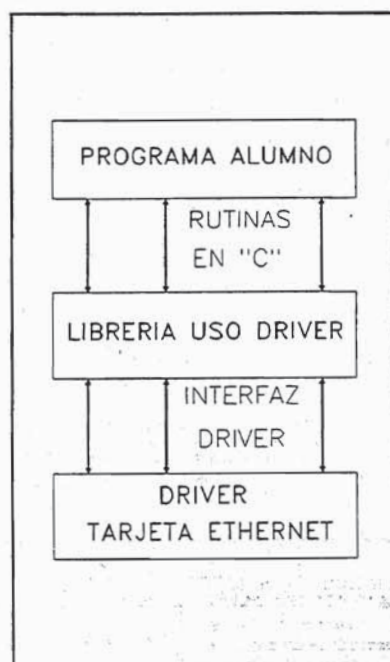


Figura 3.- Interacción con el programa del alumno.

- *enviar_paquete*. Con esta función se envían los paquetes a la red utilizando la tarjeta. Además de pasarle como parámetro los datos que se quieren enviar, es necesario indicar a quién va dirigida la trama (dirección destino) así como la longitud de la misma y el número de reintentos que se quiere realizar en caso de que se produzcan fallos.

- *recibir_paquete*. Esta función se encarga de devolver al usuario un paquete recibido por la tarjeta. Como parámetro de entrada requiere que se le indique el número de centésimas de segundo que se esperará para la recepción de un paquete en caso de no haber uno disponible.

Como puede verse el manejo de estas funciones es trivial, con solo conocer el nombre de la función y los parámetros de entrada y de salida se puede tener un control absoluto de la tarjeta ethernet.

Con esta librería se pueden realizar muchas prácticas que pueden instruir al alumno tanto en el funcionamiento de la red desde un punto de vista hardware como software. Entre las prácticas que se pueden realizar cabe destacar el realizar un programa benchmark que ofrezca una estadística de uso de la tarjeta (número de colisiones, tramas erróneas, ...), analizadores de red, diseño de routers, diseño de bridges, implementación de un protocolo [4] determinado (tcp/ip, arp, rip, ipx) etc.

5.- CONCLUSIONES

El diseño de la librería en lenguaje "C" ha servido de apoyo en la elaboración de las prácticas de la asignatura Redes de Computadores de 4º curso de la Licenciatura de Informática, ya que con esta librería se tiene una interfaz para el manejo de las tarjetas ethernet sencilla y cómoda de usar (sólo hay que conocer el nombre de la función y sus parámetros). Por otro lado cabe destacar que el alumno tiene la posibilidad de realizar las prácticas de la asignatura fuera del laboratorio, evitando de esta forma la congestión provocada por la fecha de entrega de una determinada práctica. Además cambios en el hardware -cambios de equipos y en definitiva de las tarjetas de red- no implicaría un rediseño de las prácticas, ya que la librería es independiente de la tarjeta que se utilice, o lo que es lo mismo, del fabricante.

Por último cabría destacar que el alumno no pierde la visión hardware, ya que se le enseña tanto la instalación como el funcionamiento interno de las tarjetas ethernet.

6.- REFERENCIAS

[1] David Lopez Muñoz. "Programación de la tarjeta controladora de red ethernet". Trabajo voluntario de la asignatura de Redes de Computadores, 4º curso Licenciatura de Informática, 1995.

[2] SMC, Standard Microsystems Corporation. "EtherCard Elite Ultra Adapters". User Guide, SMC, 1993.

[3] John Romkey. "PC/TCP Packet Driver Specification". FTP Software Inc, 1994.

[4] Uyless Black. "TCP/IP and Related Protocols". Mac Graw Hill, 1992