

Laboratorio de Redes. Práctica 1.

Programación con sockets raw

Objetivos

Programación de la interfaz de red a bajo nivel, en nuestro caso *ethernet*, utilizando las posibilidades que ofrece el uso de “sockets raw”. Para ello se programará una sencilla aplicación de detección de algunas propiedades básicas de los ordenadores conectados en un entorno de red local.

Desarrollo de la práctica

La utilización de los sockets en modo raw permite acceder directamente al formato del paquete enviado y recibido, por ello, es necesario conocer su estructura. En el caso de una trama ethernet el formato es el siguiente:

Trama Ethernet DIX V2.0

	Preámbulo	Dir. Destino	Dir. Origen	Tipo	Datos	FCS
Bytes	8	6	6	2	n	4

donde,

Preámbulo : es una secuencia que permite la sincronización entre el receptor y el transmisor.

Dirección Destino : Dirección física de la estación que debe recibir la trama.

Dirección Origen : Dirección física de la estación que envía la trama.

Tipo (Ethertype) : Tipo de trama que se está utilizando. Identifica la aplicación a la que pertenece de entre las múltiples que pueden estar ejecutándose simultáneamente en la misma estación.

Datos : Información que se pretende enviar.

FCS (Frame Check Sequence) : Código de redundancia cíclica (CRC) para comprobación de errores durante la transmisión.

El tamaño total de la trama debe estar entre 72 y 1526 bytes. Los campos sombreados son gestionados directamente por la tarjeta de red.

El alumno deberá realizar un programa en lenguaje C que permita al usuario conocer algunos parámetros básicos de los ordenadores conectados a la red local del laboratorio. Dicho usuario debe especificar uno de los nombres lógicos de las máquinas (alfa, beta, ...) y el programa debe pedir la siguiente información a la máquina correspondiente: Tipo de sistema operativo utilizado (puede obtenerse en el sistema con “uname -s”), versión del kernel (uname -r), el nombre de la máquina (hostname) - que debe coincidir con el nombre del ordenador especificado por el usuario-, el número de usuarios conectados y la carga media del sistema en los últimos 15 minutos.

Programas de ejemplo:

■ envia.c

```
#include <stdio.h>
#include <errno.h>
#include <unistd.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <linux/in.h>
#include <linux/if_ether.h>
#include <net/if.h>
#include <sys/ioctl.h>
#include <netpacket/packet.h>

#define ELMIO 0x1224 //Tipo de trama ethernet (distinto para cada grupo)

int main(int argc, char **argv) {
    int sock, i, fromlen;
    char buffer[2048];
    struct ifreq ifr;
    struct sockaddr_ll from;
    unsigned char destino[6]={0x00,0xE0,0x7D,0x93,0x1B,0x95}; //La conozco
    short tipo;

    //Abro el socket
    if ( (sock=socket(PF_PACKET, SOCK_RAW,
        htons(ELMIO)))<0) { //host a network (orden de los bytes)
        perror("socket");
        exit(1);
    }

    //Obtengo la direccion MAC de mi tarjeta de red
    strcpy(ifr.ifr_name, "eth0"); /* asumo que esta en eth0 */
    ioctl(sock, SIOCGIFHWADDR, &ifr); /* obtengo la MAC*/

    //la imprimo (solo por verla en pantalla)
    for(i=0; i<6; i++)
        fprintf(stderr,"%02hX:",(unsigned char) (ifr.ifr_hwaddr.sa_data)[i]);
    fprintf(stderr,"\n");

    //Preparo el paquete para enviarlo
    memcpy(buffer,destino,6); //Direccion destino
    memcpy(buffer+6,ifr.ifr_hwaddr.sa_data,6); //Direccion origen (la mia)
    tipo=htons(ELMIO); //Le doy la vuelta al short si fuese necesario
    memcpy(buffer+12,&tipo,2); //Tipo de paquete
    strcpy(buffer+14,"un mensaje que he enviado");

    //Pido algunos parametros necesarios de la estructura sockaddr_ll

    ifr.ifr_ifindex = 0;
    ioctl(sock, SIOCGIFINDEX, &ifr);

    from.sll_family = PF_PACKET;
    from.sll_ifindex= ifr.ifr_ifindex;
    fromlen = sizeof(from);
```

```

//Envio el paquete
if(sendto(sock,buffer,80,0,(struct sockaddr*) &from, fromlen)<0){
    perror("Enviando...");
    exit(1);
}

//cerramos el socket

close(sock);
}

```

■ recibe.c

```

#include <stdio.h>
#include <errno.h>
#include <unistd.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <linux/in.h>
#include <linux/if_ether.h>
#include <net/if.h>
#include <sys/ioctl.h>
#include <netpacket/packet.h>

#define ELMIO 0x1224 //Tipo de trama ethernet (distinto para cada grupo)

int main(int argc, char **argv) {
    int sock;
    char buffer[2048];

    //Abro el socket
    if ( (sock=socket(PF_PACKET, SOCK_RAW,
        htons(ELMIO)))<0) { //host a network (orden de los bytes)
        perror("socket");
        exit(1);
    }

    //Espero a que me envíen un paquete
    recvfrom(sock,buffer,2048,0,NULL,NULL);

    //Si lo recibo, lo imprimo
    fprintf(stderr,"%s\n",buffer+14);

    close(sock);
}

```