



Escuelas Superior de Cómputo

PRÁCTICA 2

APERTURA Y ENVÍO DE DATOS A TRAVÉS DE UN SOCKET NO ORIENTADOS A CONEXIÓN (UDP)

Redes de computadora

Autor:

Héctor David González Tetuán

04/11/2025

Código del programa en C

Código para abrir un socket UDP.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

int main() {
    int udp_socket, lbind, tam;
    struct sockaddr_in local, remota;
    unsigned char msj[100] = "Red, soy Tetuan";

    udp_socket = socket(AF_INET, SOCK_DGRAM, 0);

    if (udp_socket == -1) {
        perror("Error al abrir el socket");
        exit(EXIT_FAILURE);
    }

    else{

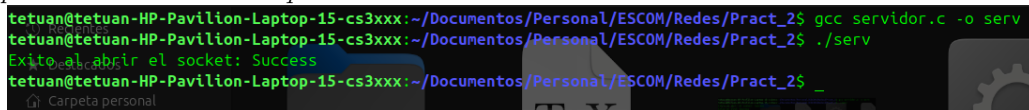
        perror("Exito al abrir el socket");

    }

    close(udp_socket);
    return 0;
}
```

Capturas de pantalla

Captuas donde se muestra que abrió un socket UDP



Código del programa en C

Código para enviar un mensaje a través de un socket UDP

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

int main() {

    int udp_socket, lbind, tam;
    struct sockaddr_in local, remota;
    unsigned char msj[100] = "Red, soy Tetuan";

    udp_socket = socket(AF_INET, SOCK_DGRAM, 0);

    if (udp_socket == -1) {
        perror("Error al abrir el socket");
        exit(EXIT_FAILURE);
    }

    else{

        perror("Exito al abrir el socket");

        local.sin_family = AF_INET;
        local.sin_port = htons(0); // Puerto de escucha
        local.sin_addr.s_addr = INADDR_ANY;

        lbind = bind(udp_socket, (struct sockaddr *)&local, sizeof(local));

        if (lbind == -1) {
            perror("Error en bind");
            exit(0);
        }

        else{

            perror("Exito en bind");

            remota.sin_family = AF_INET;
            remota.sin_port = htons(53); // Puerto DNS
            remota.sin_addr.s_addr = inet_addr("8.8.8.8");

            tam = sendto(udp_socket, msj, 20, 0, (struct sockaddr *)&remota,
                sizeof(remota));
```

```

        if (tam == -1) {
            perror("Error al enviar");
            exit(0);
        }

        else
            perror("Exito al enviar");
    }

}

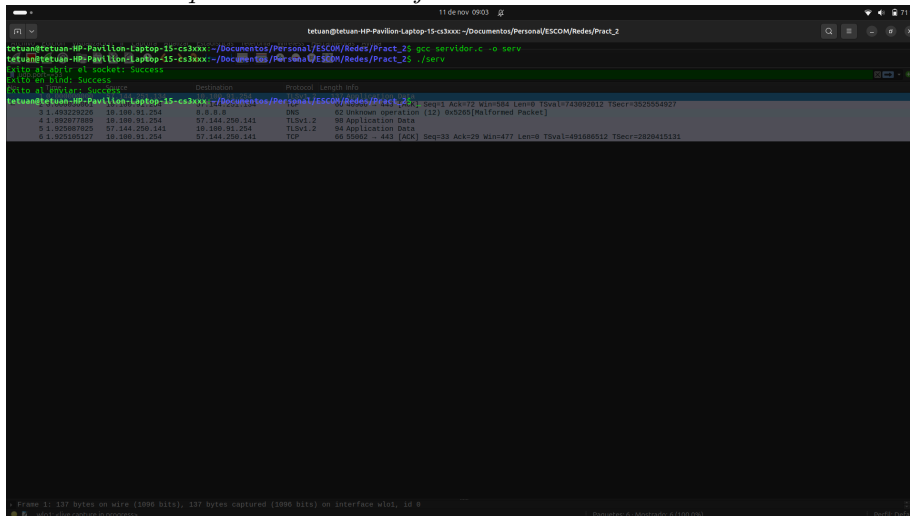
close(udp_socket);
return 0;

}

```

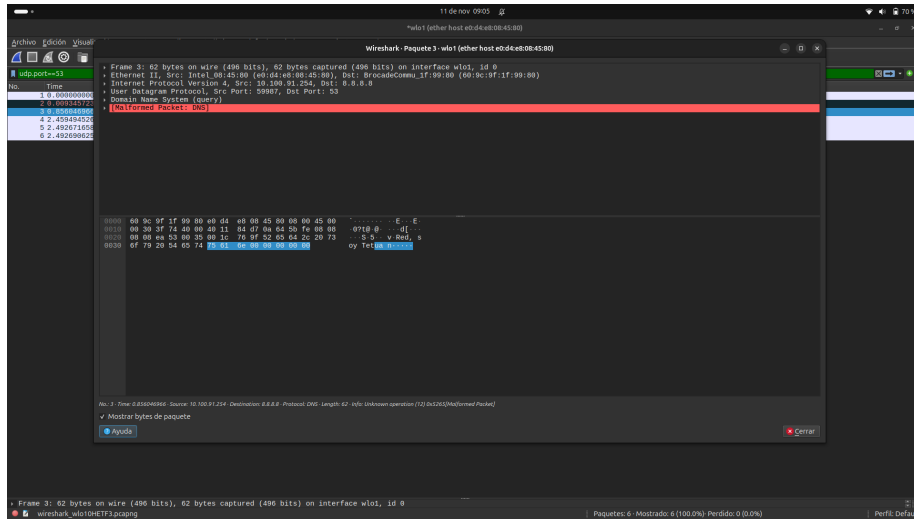
Capturas de pantalla

Donde se muestra que envió un mensaje a través de in socket UDP



Capturas de pantalla

Capturas donde se muestra que configuro el analizador de protocolos wireshark y se muestra que captura la trama enviada



Conclusión

Conocer todos los pasos para abrir un socket UDP y enviar datos a través de él, es de suma importancia en el ámbito de las redes de computadoras. UDP es un protocolo de comunicación que permite la transmisión de datos sin necesidad de establecer una conexión previa, lo que lo hace ideal para aplicaciones que requieren rapidez y eficiencia.

Además, entender cómo configurar y utilizar sockets UDP proporciona una base sólida para el desarrollo de aplicaciones de red más complejas y mejora la capacidad de los desarrolladores para optimizar el rendimiento de las comunicaciones en red.