# Práctica 5 – envio dATOS

## **Fecha: 25/10/21**

## **NOMBRE DEL EqUIPO: El Siuu team**

## **ParticipanteS:** -Fischer salazar césar eduardo

## -López García José Eduardo

## -Meza Vargas Brandon David

## **Unidad Académica: Redes de computadoras**

## **Panorámica**

## **INTRODUCCION**

La presente práctica pretende que, empleando como recursos indispensables en la creación de códigos como son los manuales de Linux, su terminal, los sockets crudos que se han estado programando desde anteriores prácticas y el software de Wireshark, se permita realizar con éxito el envío de una trama, así como observar y dar una explicación concisa de lo que se ha capturado dentro del Wireshark para considerar que la ejecución se realizó satisfactoriamente.

De igual forma, en la presente práctica se hará la estructuración correcta de una trama, para de esta forma, como se mencionó enviarla a través de la red.

Antes de seguir recordemos algo sobre las tramas.

La trama se encuentra en la capa de enlace responsable de la transmisión y separa el flujo de datos de bits en bloques o tramas. Recordemos que una trama debe tener al menos 64 bytes para que funciones la detección de colisiones y puede tener un máximo de 1518 bytes. La estructura de la trama es la siguiente:

Diagrama

Descripción generada automáticamente

Imagen . Estructura de la trama.

## **Objetivos**

# **Objetivo principal**: Programar sockets crudos y ESTRUCTURAR y ENVIA UNA TRAMA ATRAVEZ DE LA RED

# **Objetivo secundario**. ENVIAR Y capturar trama bien estructurada CON LA AYUDA DEL PROGRAMA WIRESHARK

## **Recursos necesarios para realizar la práctica**

Manuales  man socket , man 7 ip, man packet

Compilador de c

Terminal de Linux

Navegador de internet

# **Parte 1: diagrama de flujo**

Diagrama

Descripción generada automáticamente

Imagen . Diagrama de flujo de función obtenerDatos()

Diagrama

Descripción generada automáticamente

Imagen . Diagrama de flujo de función estructurarTrama()

Diagrama

Descripción generada automáticamente

Imagen . Diagrama de flujo función enviarTrama()

Diagrama

Descripción generada automáticamente

Imagen . Diagrama de flujo main()

# **Parte 2: CÓDIGOS, COMANDOS Y ejecución Y Explicación.**

* 1. INCLUIR CODIGO EXPLICANDO las estructuras del programa, Y FUNCIONES USADAS Y MENCIONAR DE QUE MANUAL DE LINUX CONSULTARON, CAMBIAR nombre de sus variables y estructuras de forma personal. Recuerden que las mejoras que le hagan al programa visto en clase aumenta su calificación.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Imagen . Código del programa para enviar tramas.

En este código se hizo una mejora, esta fue implementar un ciclo while que permita al usuario enviar tramas hasta que este ya no quiera mandar otra trama, de esta forma se evita el estar ejecutando a cada rato el programa para enviar una trama, además con esto el usuario escribe el mensaje que quiere incluir en la trama. esto nos ayudó para analizar de mejor manera las tramas enviadas usando wireshark.

* 1. CON LA AYUDA DEL wireshark capturar la trama y explicarla con capturas de pantalla,

La trama capturada fue:

Texto

Descripción generada automáticamente

Imagen . Trama capturada

Vemos que el primer dato es la mac de broadcast indicada en el programa.



Imagen . MAC broadcast

Después le sigue la dirección origen.



Imagen . Dirección origen

Le sigue el ethertype, en este caso fue el definido en el programa:



Imagen . Ethertype

Por último tenemos los datos de la trama:

Texto

Descripción generada automáticamente

Imagen . Datos trama

* 1. toda la información que reporte el wireshark.

A continuación se presenta la trama enviada y capturada por wireshark:

Imagen que contiene Calendario

Descripción generada automáticamente

Imagen . Trama enviada y capturada en wireshark

Texto, Carta

Descripción generada automáticamenteComo vemos se despliega toda la información de la trama, analicemosla primeramente por el apartado de frame:

Imagen . Apartado Frame de la trama

En esta parte primeramente nos muestra el id de nuestra interfaz, asi como el nombre, en este caso es la enp0s3 y datos generales sobre el frame como lo son:

* Tipo de encapsulamiento
* Tiempo de llegada
* La longitud del frame, en este caso es de 60, ya que así fue establecido en el código del programa

Ahora veamos la parte de ethernet ii:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Imagen . Parte de la trama Ethernet II

En esta parte se captura la dirección destino, en este caso fue la de broadcast que colocamos en el programa, de igual forma se captura la fuente, la cual como se ve, es la mac de nuestra computadora y nos muestra el ethertype.

Por último tenemos la parte de data:

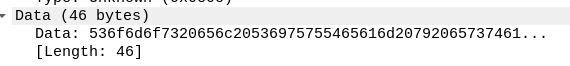


Imagen . Parte de la trama Data.

En esta parte se muestran los datos y la longitud de los datos, en este caso 46.

* 1. Incluye la captura de pantalla Y EXPLIQUE EL RESULTADO DE EJECUTAR EL PROGRAMA.

Texto

Descripción generada automáticamente

Imagen . Ejecución del programa

Al ejecutar el programa tenemos lo de la imagen anterior, primero nos solicita el nombre de la interfaz de red para obtener el índice y la MAC, posteriormente nos indica el mensaje que contendrá la trama y enviarlo.

Cuando se envía nos muestra un mensaje para volver a enviar una trama o terminar el programa.

3. conclusiones individuales de cada participante del equipo

FISCHER SALAZAR CÉSAR EDUARDO

La realización de esta práctica nos presenta un envió diferente de trama al que anteriormente habíamos utilizado, ya que en esta nosotros definimos la trama además de poder ingresar el mensaje que nosotros queremos que se envié en conjunto. Pudimos corroborar que lo hicimos él envió de información de manera correcta mediante la captura de trama para un posterior análisis de esta que efectuamos mediante el Wireshark.

Con respecto al programa anterior se me hizo un más fácil el entender cómo se hacia la consulta de datos, pero en este me costó entender un poco le definir la trama para este mensaje, después de un poco de juego con el programa se me hizo más entendible e interesante.

LÓPEZ GARCÍA JOSÉ EDUARDO

Con la ayuda de la práctica realizada, rodeando al tópico del envío de tramas, pude comprender más entendiblemente cómo es que se realizan estos envíos por medio de la red, de tal manera que se pueda definir una estructura y añadirle un pequeño mensaje. Con la ventaja que ofrece el software Wireshark, se hizo un análisis de lo que había capturado, de esta manera comprobando que se haya realizado el envío de forma correcta.

De igual manera, siento que se entendió bastante claro la manera en que se estructuran las tramas de una red, para poder checar distintos parámetros para asegurarnos de que se encuentren en forma, y que no haya ocurrido algún siniestro durante su envío.

MEZA VARGAS BRANDON DAVID

Gracias a esta práctica comprendí de mejor manera como se hace el envió de tramas por la red, pues estructuramos desde cero una trama, además de adjuntarle nuestros propios datos que fue en este caso un mensaje. Dicha trama la analizamos en wireshark para ver que la enviamos de forma correcta.

Personalmente me gustó mucho esta práctica, pues me parece muy interesante como con la función sendto, podemos enviar esta información y no solo un simple mensaje como en prácticas anteriores, sino que podemos estructurar nuestra propia trama y mandarla a través de la red. Con esto me puedo dar una idea de como el gran tráfico que pasa por la red está estructurado, en clases anteriores lo habíamos visto de manera algo teórica y con esta práctica quedo mucho más reforzado al ver nuestra propia trama.

Finalmente, creo que es importante saber cómo una trama está estructurada, pues a partir de esto podemos jugar modificando valores y ver como es enviada la trama y verificar si existió un error o algo raro.