# Práctica 2 – Mi Primer Socket

## **Fecha: 11/10/21**

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

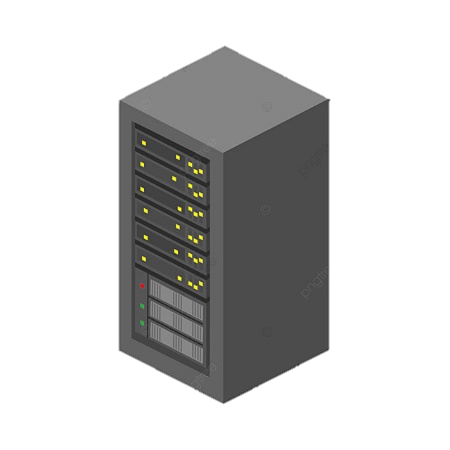
## **ParticipanteS:** -Fischer salazar césar eduardo

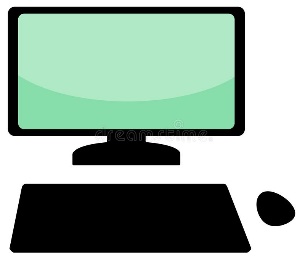
## -López García José Eduardo

## -Meza Vargas Brandon David

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

## **Panorámica**

CLIENTE SERVIDOR



Información cliente



Información servidor

Figura 01. Analogía sockets.

## **Objetivos**

# **Objetivo principal**: Programar sockets, relacionar la función bind con el socket, programar un cliente y enviar un mensaje por la red. Apoyándonos con el uso del analizador de red wireshark capturar trama, describir los encabezados de cada capa del modelo TCP/IP y capturar en imagen la trama capturada e indicar cada parte de la trama

# **Objetivo secundario**. Verificar envió de un mensaje a través de la red.

## **Escenario**

# La forma más básica de construir un sistema de comunicación es implementando la Interfaz de Programación de Aplicaciones de Socket (API de Socket - Sockets Aplicación Programming Interface).

# En esta práctica el participante deberá realizar un programa que envíe un mensaje a través de su red y poder ser capturado por medio del software wireshark.

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

* Compilador Linux y editor de Linux
* Manuales de man socket , man inet\_atom, man 2 bind, man 7 ip
* Software wireshark

# **Parte 1: diagrama de flujo**

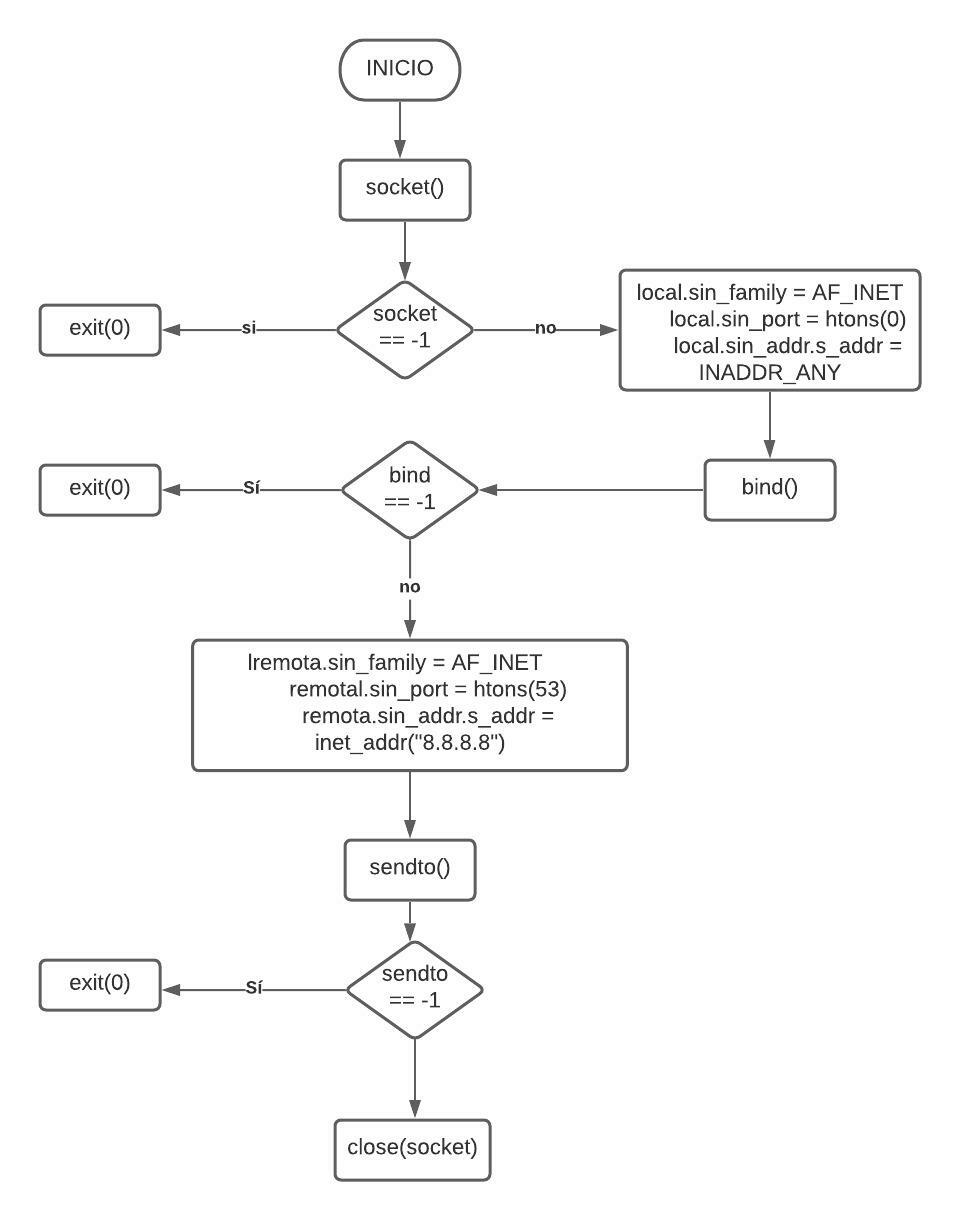


Figura 1. Diagrama de flujo del programa.

# **Parte 2: CóDIGOS, COMANDOS Y ejecución Y EXPLICACIóN.**

* 1. INCLUYE CODIGO EXPLICANDO Línea por línea hasta donde se abre el socket, el bind.

Texto

Descripción generada automáticamente

Figura 2. Código hasta la parte de bind

* 1. Incluye la captura de pantalla del resultado al ejecutar la primer parte del programa.

Texto

Descripción generada automáticamente

Figura 3. Ejecución de la primera parte

* 1. Anexa el código completo explicando líne a línea. ya con las funciones local y remota, cambiar el nombre de sus variables y estructuras de forma personal.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Figura 4. Código completo de cliente.c

* 1. ejecuta en tu terminal los comandos IFCONFIG, ARP-A, PING, PING EXTENDIDO, NSLOOKUP E INCLUYE LA CAPTURA DE PANTALLA DE CADA UNO y haz un breve comentario

ANtes de mostrar la caputra es importante hacer la siguiente instalación para tener disponibles los comandos anteriores: apt-get install net-tools

* IFCONFIG: Este comando muestra la información de las interfaces de red disponibles en la computadora:

Texto

Descripción generada automáticamente

Figura 5. Comando IFCONFIG

* ARP-A: ARP son las siglas en inglés de Address Resolution Protocol, este protocolo se encarga de encontrar la dirección hardware (Ethernet MAC) que corresponde a una dirección IP. El comando arp-a nos sirve para conocer nuestra IP, la IP y MAC de la puerta de enlace (gateway) y el tipo de direccionamiento:



Figura 6. Comando arp -a

* PING: Este comando nos sirve para comprobar si una máquina esta en red o no. Se hace un diagnostico que permite hacer una verificación del estado de una determinada conexión o host local, va seguido de una dirección:

Texto

Descripción generada automáticamente

Figura 7. Comando ping

* PING EXTENDIDO: El ping extendido realiza lo mismo que ping, con la particularidad de que este se hace de manera infinita, lo terminamos nosotros en Linux con ctrl+c:

Texto

Descripción generada automáticamente

Figura 8. Comando ping extendido

* NSLOOKUP: Para este comando necesitamos instalar dnsutils: apt-get install dnsutils. Este comando nos sirve para saber si el DNS está resolviendo correctamente los nombres e IP’s. Se usa para obtener la dirección ip conociendo el nombre y viceversa. Permite que consultemos los servidores de nombres para resolver un nombre de un host dado:

Texto

Descripción generada automáticamente

Figura 9. Comando nslookup

* 1. incluya la captura de pantalla del resultado al ejecutar el programa completo, con ayuda del programa wire shark, y explique los datos obtenidos en cada capa del modelo TCP/IP Captura y explicacion para cada capa.

Texto

Descripción generada automáticamenteEjecución completa:

Figura 10. Ejecución completa del código.

C

cAPA DE ACCESO A LA RED

En la figura que se muestra a continuación podemos ver los datos obtenidos en la capa de acceso a la red del modelo TCP/IP.

Texto

Descripción generada automáticamente

Figura 11. Datos de la capa de accesos a la red

En esta parte se captura la dirección MAC destino a donde llegó nuestro paquete enviado, además de la dirección MAC fuente que fue desde donde se mandó la información, también nos muestra el tipo, en este caso IPv4.

CAPA DE INTERNET

A continuación, podemos ver los datos que arroja wireshark capturados pertenecientes a la capa de internet.

Texto, Carta

Descripción generada automáticamente

Figura 12. Datos de la capa de transporte

En esta capa podemos ver que nos manda la versión del IP, el tamaño del cabecero, además de mostrarnos algunas banderas. De igual forma en esta capa se indica el protocolo, en este caso fue UDP.

También, al final nos muestra la dirección IP destino y la dirección IP fuente.

CAPA DE TRANSPORTE

En la siguiente figura se muestra la información de la capa de transporte arrojada por wireshark.

Texto

Descripción generada automáticamente con confianza baja

Figura 13. Información de la capa de transporte

Los datos que nos arroja es el puerto fuente que es el 33533, el puerto destino que es el 53, mismo que se especifica en el programa, además de mostrarnos la longitud.

* 1. con la ayuda de una captura de pantalla, identificar cada parte de la trama capturada.

Tabla

Descripción generada automáticamente

Figura 14. Partes de la trama.

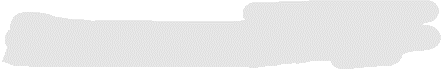


Imagen que contiene Gráfico

Descripción generada automáticamente

Imagen que contiene Gráfico

Descripción generada automáticamente

* 1. menciona graficamente y por escrito como configuraste el wireshark para recibir las tramas solo de tu tarjeta de red y hacia el servidor dns.

En primer lugar, en la parte de capture escogemos la parte de option:



Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



Figura 15. Primera configuración wireshark.

Posteriormente seleccionamos nuestra tarjeta de red y ponemos el filtro: ether host MAC como se ve en la siguiente imagen:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 16. Aplicando filtro ether host.

EN TODOS LOS CASOS AGREGA UNA BREVE EXPLICACION.

3. conclusiones individuales de cada participante del equipo

Posteriormente se da clic en start para empezar con la captura de datos.

En la siguiente pantalla, tenemos que aplicar otro filtro, donde pondremos el puerto indicado en el programa, es el puerto 53:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 17. Aplicando el filtro del puerto.



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteUna vez configurado ya podemos ejecutar el programa cliente.c para comenzar a analizar los paquetes enviados:

Figura 18. Datos capturados en wireshark.

Contesta las siguientes preguntas

**¿Qué es un socket?**

Un socket es conocido como un tipo de software que actúa como un punto final que funciona estableciendo un enlace de comunicación de red bidireccional entre el extremo del servidor y el programa receptor del [cliente](https://www.speedcheck.org/es/wiki/cliente/)

**¿Qué es la API de protocolos?**

Una API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones. Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados.

**¿Menciona brevemente como es el flujo de la Información en la arquitectura TCP/IP?**

La capa de aplicación proporciona a las aplicaciones la capacidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos. La capa de transporte de esta arquitectura es responsable de proporcionar a la capa de aplicación, servicios de sesión y de comunicación de datagramas. La capa de Internet es responsable de las funciones de direccionamiento, empaquetado y enrutamiento. Y la capa de interfaz de red es responsable de la colocación de paquetes TCP/IP en la red y de la recepción de paquetes TCP/IP de fuera la red.

**¿Menciona el nombre que se le va dando a la información en cada capa y sus encabezados?**

Cabecera TCP

Datos de capa superior

Segmento

Transporte

Cabecera IP

Datos

Paquete

Red

Cabecera LLC

Datos

Trama

Enlace de datos

Cabecera MAC

Datos

Bits

Física

0101110101001000010

**¿Qué es un Socket?**

Un socket es un método para la comunicación entre un programa del cliente y un programa del [servidor](https://www.ecured.cu/Servidor) en una [red](https://www.ecured.cu/Red), se define, por tanto, como el punto final en una conexión.

**¿Qué es un DNS?**

El sistema de nombres de dominio (DNS) es el directorio telefónico de Internet. Las personas acceden a información en línea mediante nombres de dominio (como espn.com). El DNS traduce los nombres de dominio a direcciones IP para que los navegadores puedan cargar los recursos de Internet.

**¿Qué es una dirección Lógica?**

Es aquella dirección generada por la **CPU** mientras un programa se está ejecutando; tiene la característica de ser virtual, ya que no existe físicamente. Esta dirección se utiliza como referencia para acceder a la ubicación de la memoria física

**¿Qué es una dirección Física?**

**La dirección física** identifica una ubicación física en una memoria. MMU (**Unidad de gestión de memoria)**calcula la dirección física para la dirección lógica correspondiente. MMU también utiliza la dirección física de computación de direcciones lógicas.

**¿Qué es La NIC?**

Una NIC o tarjeta de interfaz de red es un dispositivo que conecta físicamente una computadora a una red. Esta conexión permite la comunicación de alta velocidad a las impresoras, enrutadores, computadoras u otros módems de banda ancha. Los tipos más comunes de tarjetas de red incluyen tarjetas Ethernet, inalámbricas y red en anillo.

**¿Explique brevemente como se hace la consulta de un cliente a un servidor a través de la Red?**

El cliente abre un socket para poder realizar esta conexión, posteriormente se realiza un bind, si el bind se realiza correctamente, el cliente procede a preguntar a una dirección remota si es posible mandarle datos, por su parte el servidor recibe esos datos. Todo esto usando el modelo OI, sabiendo que las capas superiores utilizan los servicios de las capas inferiores para ir generando el encapsulamiento de datos, esto lo vemos en nuestro programa con las distintas funciones que usamos.

3. conclusiones individuales de cada participante del equipo

FISCHER SALAZAR CÉSAR EDUARDO

Siendo la primera vez que tenía contacto con la programación de un socket esta práctica me ayudo a entender el cómo funcionan y en que capa se trabajan, ya que conforme fuimos avanzando pudimos construir que nos permitirá abrir un socket y modificarlo para que este nos permitiera enviar un paquete del cual logramos observar su contenido gracias a la captura de su trama que realizamos en wireshark.

LÓPEZ GARCÍA JOSÉ EDUARDO

Esta práctica nos ha ayudado a comprender y conocer mucho acerca del manual que ofrece Linux para la programación de sockets, ver cómo es su funcionamiento, qué elementos requiere para que operara de la mejor manera; y gracias al uso de la herramienta de wireshark, se logró comprender y visualizar toda la información que arrojó la apertura del socket dentro de la terminal del software.

MEZA VARGAS BRANDON DAVID

Gracias a esta práctica comprendí de mejor manera lo visto anteriormente en clase sobre sockets, las capas donde trabajan, así mismo como aprender a identificar los valores o información capturada en la trama al enviar un paquete, esto con la ayuda de wireshark, programa del cual no tenía conocimiento anteriormente, pero gracias a esta práctica logre hacer uso de este.

De igual forma, con esta práctica hacemos un acercamiento a lo que es la programación de sockets y nos damos la idea de cómo programas más grandes que involucran la red, se hacen.

Una práctica de suma importancia ya que es la base de todo lo que seguiremos viendo a lo largo del semestre.

# Bibliografía

1. Barcina, F. (2020). “Comandos de red de net-tools”. Obtenido de: <https://delarioja.org/articulos/category-blog/134-comparativa-de-comandos-de-red-de-net-tools-y-iproute2>
2. [¿Qué es un socket? (speedcheck.org)](https://www.speedcheck.org/es/wiki/socket/)
3. [¿Qué es tarjeta de red o NIC? Definición, función y tipos de NIC | FS comunidad](https://community.fs.com/es/blog/nic-card-guide-for-beginners-functions-types-and-selection-tips.html)
4. [Diferencias entre Dirección Física y Dirección Lógica (pc-solucion.es)](https://pc-solucion.es/2018/04/17/diferencias-entre-direccion-fisica-y-direccion-logica/)