PRÁCTICA 8 – PROTOCOLO ARP

## **Fecha: 25/11/2021**

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

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

## -López García José Eduardo

## -Meza Vargas Brandon David

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

**INTRODUCCION**

Como se ha venido realizando desde prácticas anteriores, se ha estado manejando el empleo de los sockets crudos en Linux, así como contar con la ayuda del manual que está incluido en este sistema operativo, con el propósito de poder codificar programas que nos permitan enviar y recibir tramas situados en la red o en nuestro propio dispositivo.

En esta ocasión, estaremos visualizando con la herramienta de Wireshark el envío y llegada de tramas que incluyen el protocolo ARP, el cual se ha mencionado con anterioridad que funciona cuando se asignan direcciones IP usando este protocolo y distingue si la IP del host de destino se encuentra en la misma subred o en otra, esto para que pueda llevar a cabo una revisión de la máscara de la subred. Así, con los recursos que ya se han generado a través de las prácticas anteriores, y los ajustes solicitados para mostrar el cambio de color en las tramas, intentaremos observar a detalle cada una de las tramas que llegan al ordenador y poder darle la interpretación correcta a dichos elementos capturados.

OBJETIVOS

RECURSOS NECESARIOS PARA REALIZAR LA PRÁCTICA

**OBJETIVO PRINCIPAL**: ENVIAR UNA TRAMA ARP Y CAPTURARLA CON WIRE SHARK LINUX

**OBJETIVO SECUNDARIO**. IMPRIMIR LA TRAMA ARP ( SOLICITUD Y RESPUESTA) EN CONSOLA E INTERPRETAR LAS TRAMAS QUE RECIBIMOS CON EL PROGRAMA

Manuales  man socket , man 7 ip, man packet

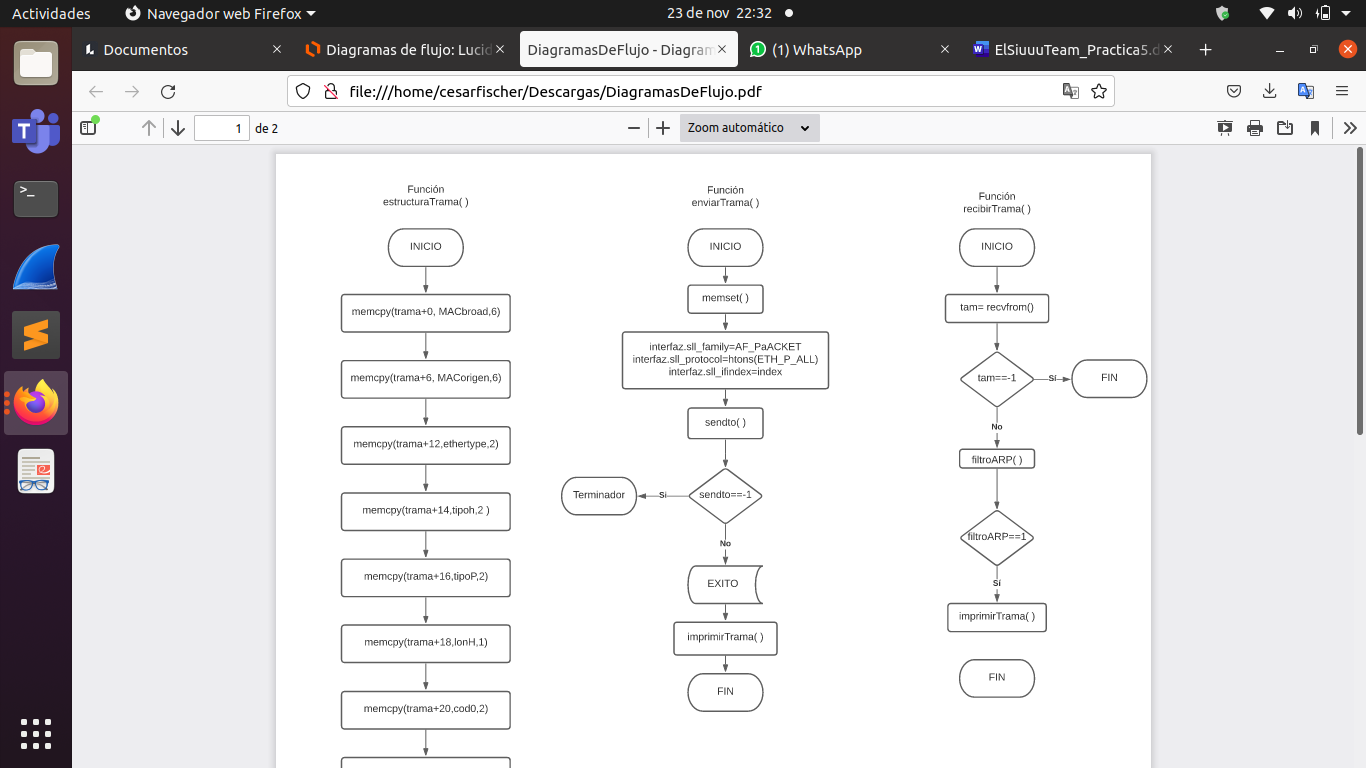
Compilador de c

Terminal de Linux

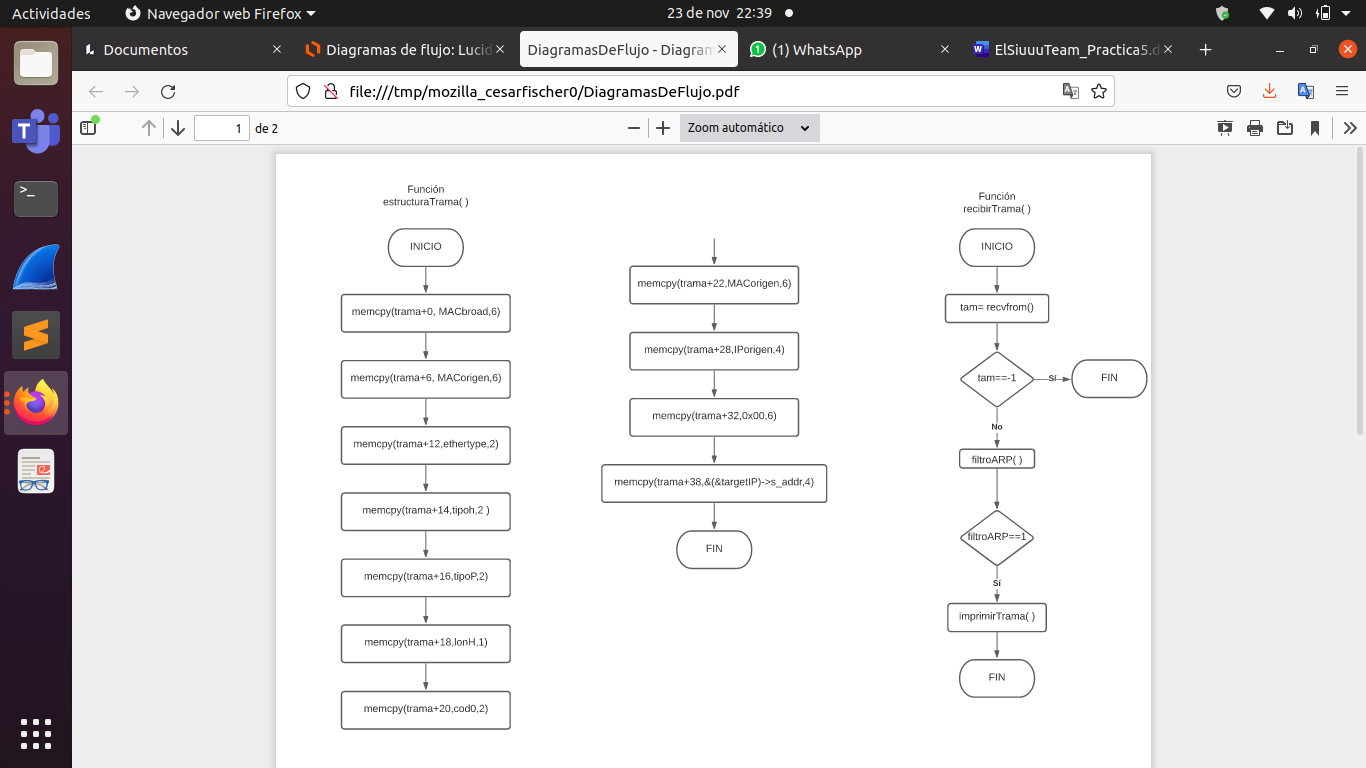
Navegador de internet

**PARTE 1: DIAGRAMA DE FLUJO**

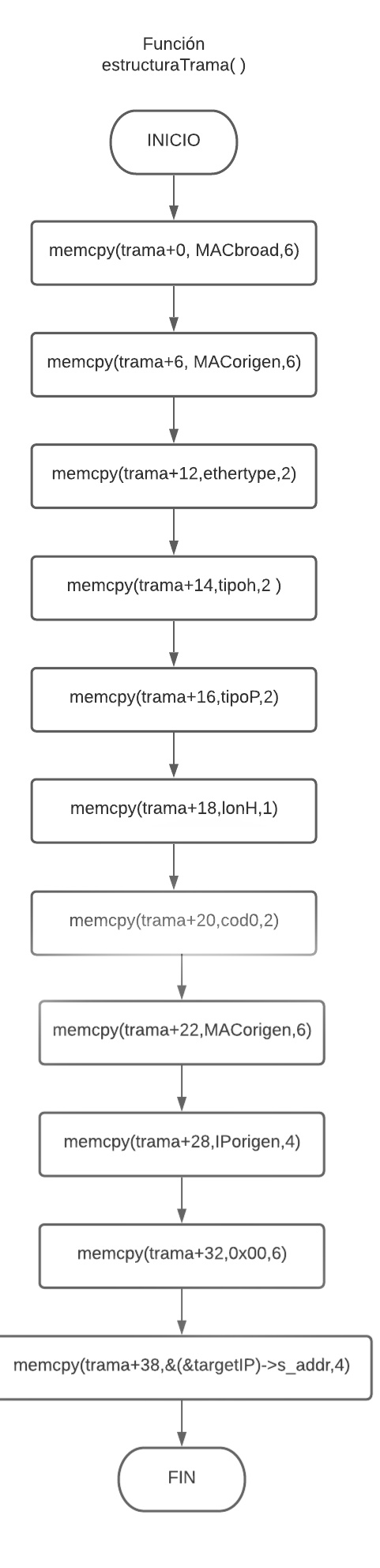
* **INCLUYE DIAGRAMA DE FUJO DEL PROGRAMA**



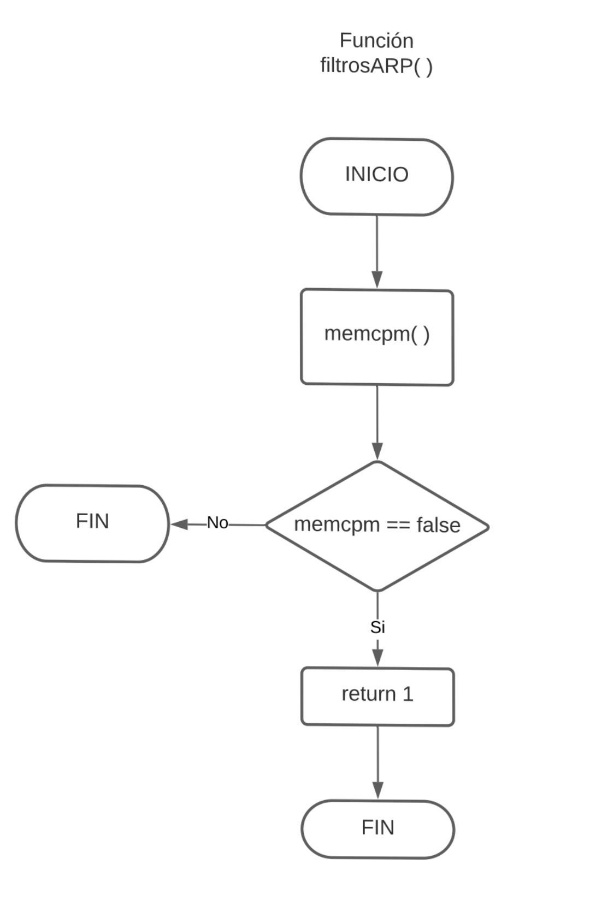
**Imagen 1. Diagrama de flujo de función enviarTrama( )**



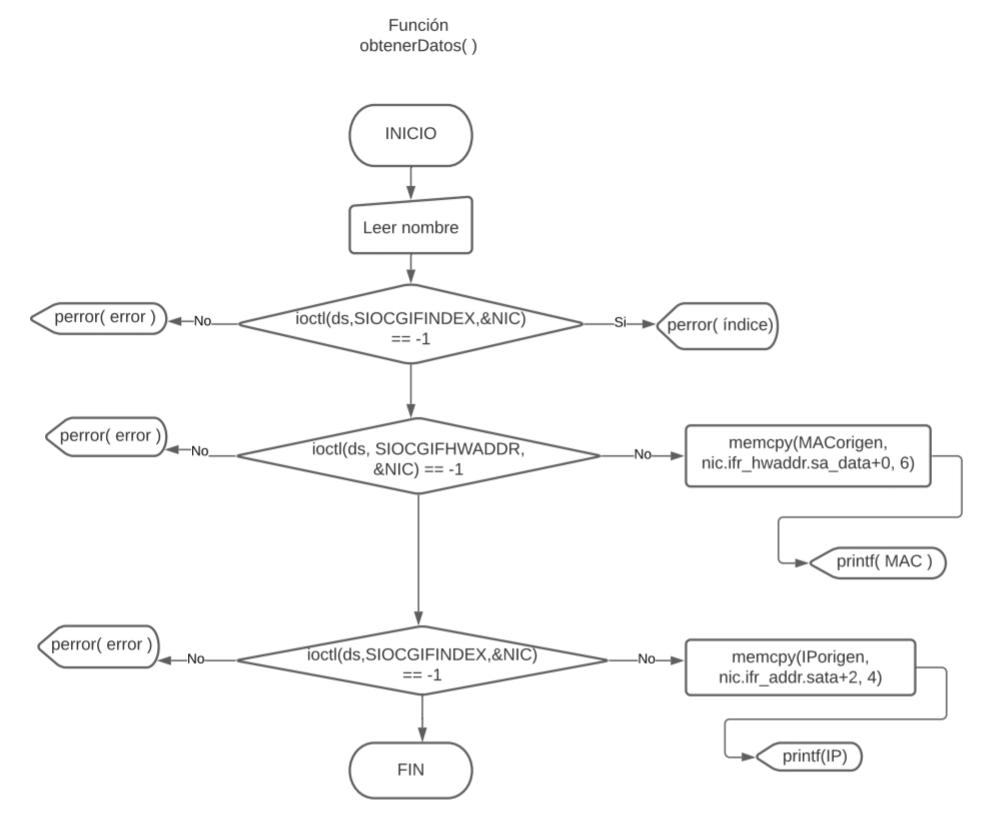
**Imagen 2. Diagrama de flujo de función recibirTrama( )**



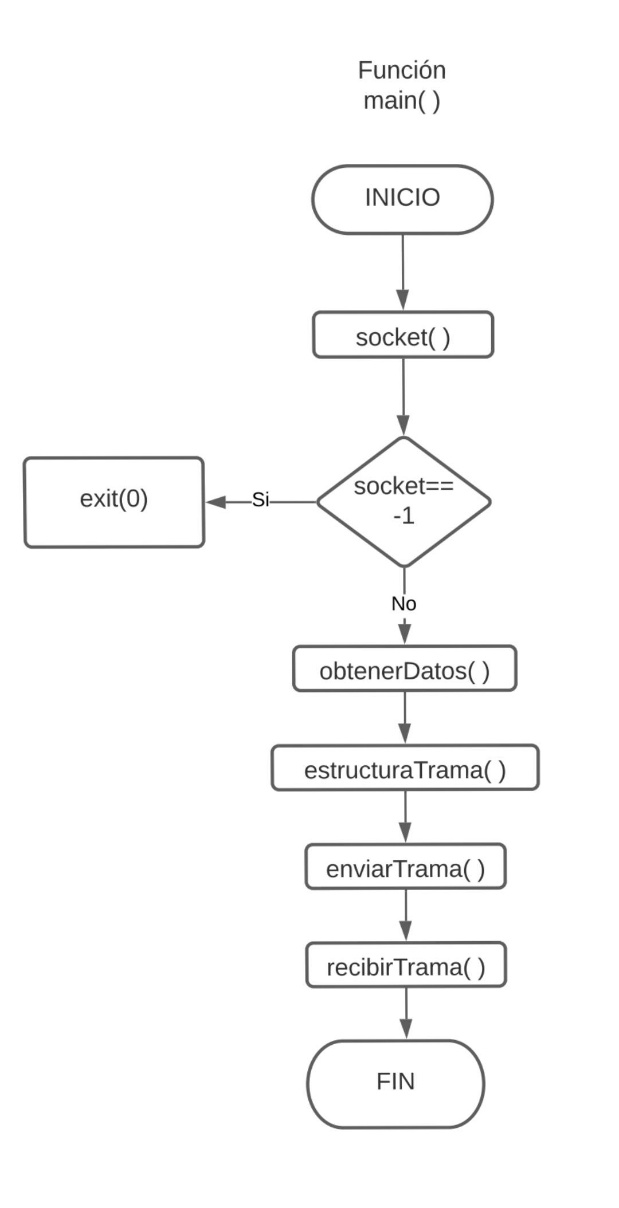
**Imagen 3.Diagrama de flujo de función estructuraTrama( )**



**Imagen 4.Diagrama de flujo de función filtros ARP( )**

****

**Imagen 5. Diagrama de flujo de función obtenerDatos()**

****

**Imagen 6.Diagrama de flujo de función estructuraTrama( )**

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. (LINEA ALINEA)**

A continuación, vemos las capturas del código utilizado comentado y explicado línea por línea.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

* 1. **EJECUTAR TOMAR CAPTURA DE PANTALLA DE CADA ETAPA DEL PROGRAMA.**

Primeramente, se puede ver la parte donde el programa obtiene los datos de nuestra interfaz de red a partir del nombre de nuestra interfaz y espera a que el usuario ingrese una IP destino.

Texto

Descripción generada automáticamente

Al ingresar una IP, se imprime la trama enviada identificando cada parte de la trama por colores especificando cada parte en texto.

Texto

Descripción generada automáticamente

Posteriormente se imprime la trama recibida que será la respuesta de ARP obtenida con el mismo formato que la trama enviada

Imagen que contiene Texto

Descripción generada automáticamente

* 1. **INCLUYE LA CAPTURA DE PANTALLA DE LAS TRAMAS Y EXPLIQUE LOS DATOS RECIBIDOS EN CADA PARTE DEL PROGRAMA**

**Trama enviada**

Texto

Descripción generada automáticamente

Primeramente, se imprime la trama enviada, esta es una petición de broadcast, esto lo podemos saber por la dirección de broadcast al inicio, posteriormente de la dirección de broadcast se imprime la MAC origen, en este caso la de nuestra computadora, posteriormente sigue el código de operación, en este caso es 0806, el cual indica que Estanis haciendo uso de ARP, después va el tipo de ethernet que es IP, seguido del tipo de protocolo, después siguen las longitudes de direcciones y de protocolo. después vuelve a aparecer nuestra dirección MAC origen seguida de la IP origen, finalmente aparece la MAC destino en este caso no hay y la IP destino que es la que ingreso el usuario.

**Trama recibida:**

Texto

Descripción generada automáticamente

La trama recibida tiene el mismo formato anteriormente explicado, el detalle aquí es que al inicio aparece la MAC origen de la respuesta, siendo esta nuestra MAC. de igual forma cambia nuestra MAC origen, además de que ahora la IP origen se convierte en nuestra IP destino y viceversa.

* 1. **MODIFIQUE EL PROGRAMA PARA CAMBIAR DE COLOR CADA PARTE DE LA TRAMA (SOLICITUD Y RESPUESTA) QUE RECIBIMOS AGREGUE CAPTURAS DE PANTALLA DE LA TRAMA CAPTURADA Y EXPLIQUE LOS DATOS.**

Estos cambios ya se realizaron anteriormente, las capturas mostradas anteriormente muestran cómo se realizó y los resultados de las tramas y los datos.

3. CONCLUSIONES INDIVIDUALES DE CADA PARTICIPANTE DEL EQUIPO

FISCHER SALAZAR CÉSAR EDUARDO

Gracias a la implementación del código de la presente practica pude comprender claramente el funcionamiento del protocolo ARP, ya que pudimos realizar una consulta mediante la estructuración de una trama la cual nos permitiera conocer la MAC perteneciente a la dirección IP indicada dentro de nuestra red de área local, dicha trama fue identificada con un código de color que nos permitiera distinguir los elementos importantes de los cuales está conformada, tales como la IP de origen y de destino , la dirección MAC de destino y el tipo de protocolo que fue utilizado. Una vez terminada la ejecución del programa pudimos verificar su funcionamiento correcto mediante la herramienta wireshark la cual nos mostró que los datos de la consulta y de la respuesta eran los mismo a los que nos arrojaba el programa.

Puedo concluir que esta práctica fue algo interesante ya que nos permite conocer más sobre el cómo funciona la comunicación entre nuestras computadoras.

LÓPEZ GARCÍA JOSÉ EDUARDO

Por medio de la implementación de este código, pudimos darnos cuenta de que esta práctica del protocolo ARP tiene una similitud/unión considerable de lo que hemos visto en las anteriores prácticas de envío y recibo de tramas de red, ahí se han empleado los mismos recursos del manual, el wireshark y los sockets para poder ver la trama enviada o recibida, con el detalle que esta corresponde a una respuesta del protocolo ARP. Aquí la implementación de los colores en la identificación de elementos de la trama ayudó a que pudieran distinguirse unos de otros, si era la MAC, la IP de origen o destino, del hardware, o si se trataban de bits de relleno.

Por tanto, puede decir que he comprendido de forma más clara cómo es que el ARP interviene en la resolución de las direcciones (tal como su nombre en inglés dice), para que así ayude a la vinculación de la IP con la MAC y exista un reconocimiento entre los sistemas.

MEZA VARGAS BRANDON DAVID

Esta práctica fue la combinación de dos prácticas anteriores, la de envió de una trama y la de recibir una trama, en esta hubo una particularidad, pues la trama enviada fue estructurada como una petición ARP, por lo tanto, la trama recibida fue una respuesta de ARP.

Gracias a esta práctica logré comprender de mejor forma el total funcionamiento del protocolo ARP, pues al estructurar una petición de ARP los conceptos que hemos visto a lo largo de las clases quedaron entendidos y aterrizados en la práctica con la realización del programa de la práctica actual.

Concluyendo así que la importancia del protocolo ARP radica en asociar dirección IP a una dirección MAC que se reconozca en la red local, en este caso de nuestra computadora y nuestra red de internet.