

Documentación proyecto

Comunicaciones y Redes

Joan Sebastián Velandia (Author)

Ingeniería de Sistemas
Pontificia Universidad Javeriana
Bogotá, Colombia

Gustavo Antonio Rivera (Author)

Ingeniería de Sistemas
Pontificia Universidad Javeriana
Bogotá, Colombia

Santos David Nuñez (Author)

Ingeniería de Sistemas
Pontificia Universidad Javeriana
Bogotá, Colombia

Abstract—*This is to let to know documentation of the program that will emulate protocol ARP into Ethernet frames. This document is compounded of objectives, definition of the classes of the program.*

Keywords—*Frames, ARP, protocol, LAN, host, IP, MAC, heredar, implementar, GUI.*

I. OBJETIVOS

Entender el comportamiento de las capas de enlace y red.

Emular el protocolo ARP.

Definir estados de máquinas host a partir del protocolo ARP.

II. FUNCIONALIDAD DEL PROGRAMA

A partir del comportamiento de las capas de enlace y red, el programa busca emular el protocolo ARP, generando sus respectivas tramas con el propósito de monitorear una red LAN y así poder identificar y clasificar los diferentes dispositivos que se encuentren conectados en esta red, de tal manera que en una primera ejecución del programa se realizan múltiples envíos de tramas hacia los posibles Host que pudieran existir en la red LAN para verificar el estado de los mismos.

Una vez encontrados los Host activos, se continúa enviando tramas con el fin de saber si estos Host aún se encuentran activos o no, dándoles un tipo de prórroga para que responda determinada cantidad de tramas, valores que pueden ser manipulados por el usuario del programa. Para mostrar el anterior proceso se emplea una interfaz amigable para el usuario donde se visualiza la finalidad del protocolo ARP.

III. DEFINICIÓN DE CLASES

Para llevar a cabo este proyecto se empleó el lenguaje de programación de java, implementando un conjunto de librerías entre las cuales se encuentra la librería jpcap la cual permite

generar, enviar y recibir diferentes paquetes, donde manejamos específicamente el tipo de tramas ARP, además de ello, esta librería contiene funcionalidades que incluyen la obtención de direcciones IP y MAC, a través de sus clases y métodos preestablecidos.

Las clases que se desarrollaron se clasificaron en base a la arquitectura de software MVC (modelo vista controlador), distribuidas como se muestra a continuación:

Modelo:

Las clases que encontramos en esta sección son:

- ARP: Contiene toda la información relacionada a con la creación tramas ARP.
- Dispositivo: Representa un host conectado en la red LAN y su información como: estado actual, el tipo de dispositivo, IP, MAC, entre otros.
- TipoDispositivos: Clase enumerado que contiene los diferentes tipos de dispositivos que pueden estar conectados en la red.
- TiposEstado: Clase enumerado que contiene los posibles estados del dispositivo (Activo, Pendiente o Inactivo) que se encuentre en la red monitoreada.
- IEmulador: Interfaz que contiene los principales métodos para que el programa pueda emular correctamente el protocolo ARP.
- Emulador: Clase que implementa la interfaz IEmulador y describe como deben establecerse los principales métodos para que el programa pueda emular correctamente el protocolo ARP.

Vista:

Este paquete contiene las siguientes clases:

- InterfazGrafica: Clase que hereda de JFrame, contiene la ventana del programa con al que interactúa el usuario y da acceso a los diferentes paneles que se encuentran en este paquete (MVC).

- **ImagenesT:** Clase que hereda de la clase `DefaultTableCellRenderer`, y es empleada por la clase `PanelDispositivos` para la modificación de la tabla que muestra al usuario el monitoreo de la red.
- **PanelMenu:** Es el primer Panel que se muestra cuyo objetivo es obtener la información de red LAN proporcionada por el usuario y empleada para realizar el proceso de búsqueda de los dispositivos
- **PanelDispositivos:** Panel que genera la tabla donde se muestran los dispositivos conectados en la red y sus características. Recoge información sobre el tiempo de espera y cantidad de veces que se envía una trama permitiendo actualizar el estado de los dispositivos.

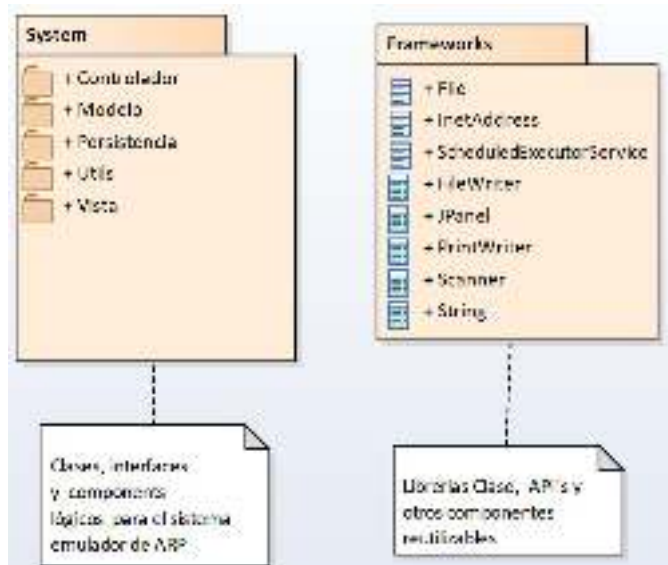
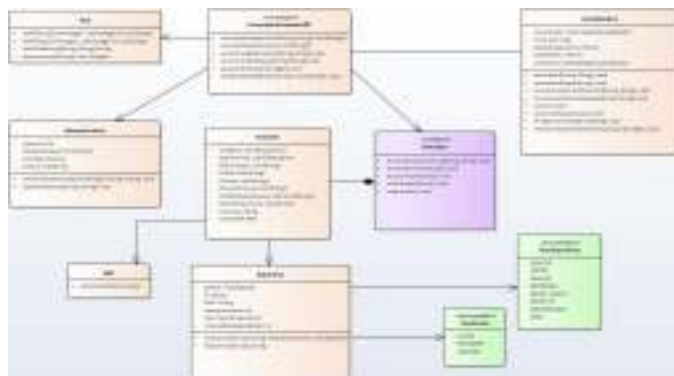
Controlador:

Para este paquete tenemos una única clase llamada `ControladorEmuladorARP` el cual se encarga de responder a los eventos GUI ejecutando acciones en base a dichas entradas.

También se usan otros dos paquetes. El paquete de persistencia, el cual contiene la clase `ManejoArchivos` la cual genera la persistencia de los datos de los dispositivos a los que se acceden durante el proceso.

El paquete de utils el cual posee una única clase, `Utils`, la cual implementa una serie de métodos que proporcionan herramientas para facilitar el manejo de la información, durante el proceso.

A continuación se presenta el diagrama UML del programa:



IV. PROTOCOLOS

A. Protocolo de resolución de direcciones Ethernet

La implementación del protocolo P en un host S emisor decide, a través del mecanismo de enrutamiento del protocolo P, que quiere transmitir a un host objetivo T ubicado en algún lugar en una pieza conectada de Cable Ethernet 10Mbit. Para transmitir realmente el paquete Ethernet. Se debe generar una dirección Ethernet de 48.bit. Las direcciones de los hosts dentro del protocolo P no siempre son compatibles con el Dirección Ethernet correspondiente (siendo diferentes longitudes o valores). Aquí se presenta un protocolo que permite la dinámica distribución de la información necesaria para construir tablas para traducir una dirección A en el espacio de direcciones del protocolo P en una Dirección Ethernet 48.bit.

Se han hecho generalizaciones que permiten que el protocolo sea Se utiliza para hardware Ethernet que no es de 10Mbit. Algunos paquetes de radio. Las redes son ejemplos de dicho hardware.[1]

B. Trama ethernet II

Campos Preámbulo y Delimitador de inicio de trama:
Los campos Preámbulo (7 bytes) y Delimitador de inicio de trama (SFD) (1 byte) se utilizan para la sincronización entre los dispositivos emisores y receptores. Estos ocho primeros bytes de la trama se utilizan para captar la atención de los nodos receptores. Básicamente, los primeros bytes le indican al receptor que se prepare para recibir una trama nueva.

Campo Dirección MAC de destino. El campo Dirección MAC de destino (6 bytes) es el identificador del receptor deseado. La dirección de la trama se compara con la dirección MAC del dispositivo. Si coinciden, el dispositivo acepta la trama.

Campo Dirección MAC de origen. El campo Dirección MAC de origen (6 bytes) identifica la NIC o interfaz de origen de la trama.

Campo Longitud/tipo. Para todos los estándares IEEE 802.3 anteriores a 1997, el campo Longitud define la longitud exacta del campo de datos de la trama. Esto se utiliza posteriormente como parte de la FCS para garantizar que el mensaje se reciba adecuadamente. Si el objetivo de un campo es designar un tipo como en Ethernet II, el campo Tipo describe cuál es el protocolo que se implementa.

Campos Datos y Pad. Los campos Datos y Pad (de 46 a 1500 bytes) contienen los datos encapsulados de una capa superior, que es una PDU de Capa 3 genérica o, con mayor frecuencia, un paquete IPv4. Todas las tramas deben tener al menos 64 bytes de longitud. Si se encapsula un paquete pequeño, el Pad se utiliza para incrementar el tamaño de la trama hasta alcanzar el tamaño mínimo.

Campo Secuencia de verificación de trama. El campo Secuencia de verificación de trama (FCS) (4 bytes) se utiliza para detectar errores en la trama. Utiliza una comprobación cíclica de redundancia (CRC). El dispositivo emisor incluye los resultados de una CRC en el campo FCS de la trama.[2]

V. PROTOCOLO DE PRUEBAS

Protocolo de Pruebas

ESCENARIO I

En este escenario se encuentran cuatro dispositivos(PCs) conectados a la red LAN, después de un tiempo uno de los dispositivos se desconecta.

El usuario ingresa los siguientes campos al programa (Panel Menú)

IP Host: 192.168.0.1
Mascara de Red: 255.255.255.246

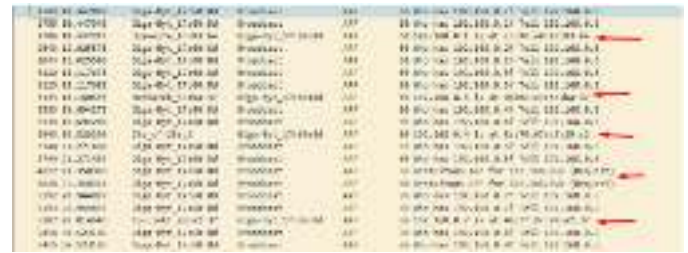
donde la IP de la máquina que ejecuta el programa es: 192.168.0.6

El programa ejecuta el envío de tramas ARP en modo broadcast, permitiendo un tiempo límite de escucha de 1.5 segundos a los posibles host (192.168.0.1 - 192.168.0.10).

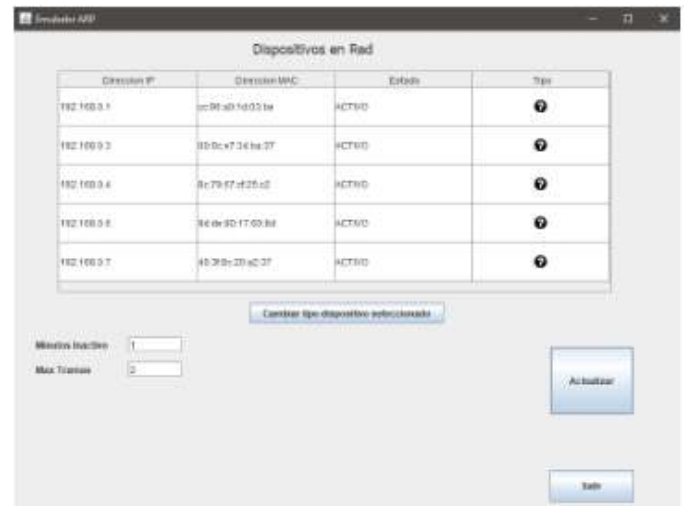
Donde son respondidas por los hosts: 192.168.0.1(Modem de la red)

192.168.0.3(Host 1)
192.168.0.4(Host 2)
192.168.0.6(Host 3)
192.168.0.7(Host 4)

Tal como se muestra gracias a la herramienta WireShark:

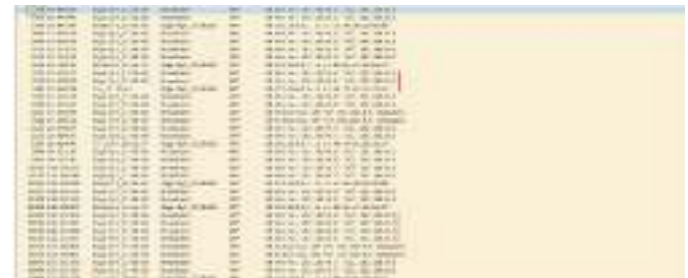


Mostrándose en el panel dispositivos:

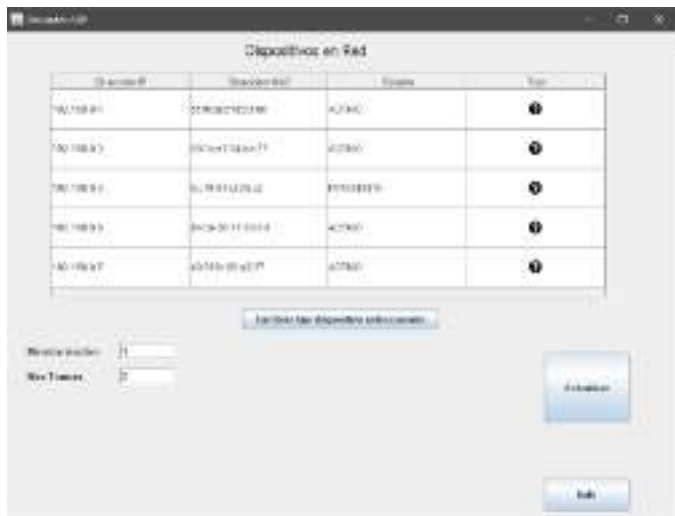


Continuamos con una actualización donde se desconectan los dispositivos y se cambia el tipo de dispositivos.

Encontrando que el host con IP 192.168.0.4 no respondió el llamado.



El Panel dispositivos procede a tomarlo como inactivo.



A continuación se espera la actualización automática donde este host no responde a los llamados y pasa a estado Inactivo.



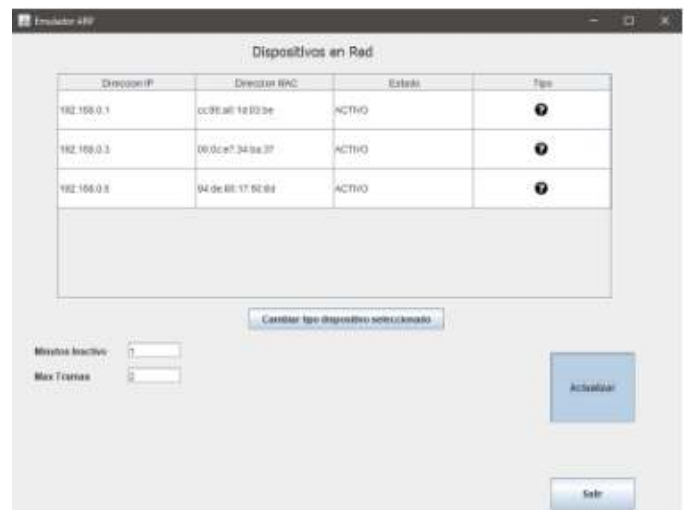
Se procede a salir del programa, guardando la sesión actual.



ESCENARIO II

Se realiza el escenario I, con la diferencia que ahora la sesión ha sido guardada, teniendo esto en cuenta el programa revisa la base de datos y envía tramas ARP únicamente los host que allí se encuentran. Se muestra el mismo panel con el que finalizo el escenario I.

Se procede a actualizar manteniendo el host 2 y host 4 fuera de la Red obteniendo.



REFERENCES

- [1] David C. Plumer. AN INTERNET ADDRESS RESOLUTION PROTOCOL online. Disponible: <https://tools.ietf.org/html/rfc826>
- [2] IEEE ESTÁNDAR FOR ETHERNET online. Disponible <https://standards.ieee.org/about/get/802/802.3.html>