

Proyecto 1: Identificación de red

PRESENTADO POR:
Juan Manuel Sánchez
Stephanie Domínguez

PROFESOR:
Enrique Ruiz



Pontificia Universidad Javeriana
Facultad de Ingeniería
Ingeniería de Sistemas
Comunicación y Redes

Bogota D.C
25 de Septiembre 2016

Introducción

El presente proyecto pretende aplicar los conocimientos adquiridos en la clase de comunicación y redes con el objetivo de emular el funcionamiento del protocolo ARP, por otro lado se pretende hacer uso de los conocimientos de los protocolos de la capa de red y de enlace.

Pasos para utilizar el programa:

1. El usuario ejecuta el programa.
2. El usuario observa un menú donde se encuentran los siguientes ítems:
 - a. **Dirección IP:** aparece la dirección de la máquina en la cual se encuentra ubicado el usuario.
 - b. **Mascara de red:** Se muestra la máscara que utiliza esa dirección.
 - c. **Timeout para que la máquina responda:** en este espacio el usuario puede definir cuanto tiempo en milisegundos puede demorar una máquina para responder un mensaje.
 - d. Numero de mensajes para poder cambiar el estado:
 - i. Pendiente: El usuario puede modificar el total de mensajes que hay que enviar a una máquina para que pase a un estado activo a pendiente.
 - ii. Inactivo: El usuario puede modificar el total de mensajes que hay que enviar a una máquina para que pase a un estado pendiente a inactivo.
 - e. El usuario puede modificar el tiempo en segundos para cuando una máquina este en estado inactivo sea eliminada del sistema.
3. El usuario debe escoger de qué manera debe identificar la red:
 - a. Automática:
 - i. El sistema debe mandar un broadcast a todas las máquinas.
 - ii. Las máquinas retornan la Mac
 - iii. El sistema las identifica, las pone en estado activo y aumenta el número de mensajes enviados y de mensajes respondidos.
 - iv. El sistema muestra:
 1. El tipo de dispositivo.
 2. La dirección IP de la máquina.
 3. La dirección MAC de la máquina.
 4. El estado de la máquina (ACTIVO, PENDIENTE, INACTIVO).
 5. Mensajes enviados: El número de mensajes ARP que han sido mandados a esta máquina.
 6. Mensajes perdidos: El número de mensajes que no respondió la máquina
 - v. El sistema espera 10 segundos antes de mandar un nuevo mensaje broadcast.
 - vi. El sistema verifica que máquinas ya fueron identificadas previamente (paso iii) y aumenta su número de mensajes enviados y respondidos o dado el caso aumenta los mensajes perdidos.

- vii. En caso de que antes no se encontrara una maquina el sistema la identificara y la colocara para que el usuario la vea, se repetirá el paso iii.
- viii. Se repite este proceso hasta que el usuario seleccione manual o cierre el programa.
- b. Manual:
 - i. El usuario debe seleccionar el botón enviar ARP
 - ii. El sistema dependiendo del timeout que usuario introdujo este va enviar un mensaje ARP a diferentes máquinas.
 - iii. Para cada máquina identificada se crea un hilo el cual le manda cada 5 segundos un mensaje para identificar su estado este puede cambiar de ACTIVO a PENDIENTE y por ultimo a INACTIVO, este cambio varia con respecto a los mensajes que el usuario indico (**ver paso d**) .
 - iv. El sistema muestra cada maquina identificada con los siguientes ítems:
 - 1. El tipo de dispositivo.
 - 2. La dirección IP de la máquina.
 - 3. La dirección MAC de la máquina.
 - 4. El estado de la maquina (ACTIVO, PENDIENTE, INACTIVO).
 - 5. Mensajes enviados: El número de mensajes ARP que han sido mandas a esta máquina.
 - 6. Mensajes perdidos: El número de mensajes que no respondió la maquina
 - v. Se repite este proceso hasta terminar la totalidad de las máquinas.

EXEPERIMENTO:

Para este proyecto empezamos realizando pruebas en una casa de uno de los integrantes del grupo, conenctando un computador portatil al router y reconociendo los dispositivos del area local, por ejemplo los celulares y otras maquinas de la casa. Despues de esta primera prueba, se realizaron experimentos en la sala de redes y nos dimos cuenta que la versión del eclipse (Helios) no fnciono el proyecto ya que una de las librerias no funciono de manera adecuada. Al principio se encontraron errores en el proyecto ya que no se borraban los dispositivos que se desconectaban del HUB, como consecuencia se corrigio este error en una nueva versión. Por ultimo probamos con un switch y funciono de manera adecuada.

RESTRICCIONES:

1. **Eclipese mars:** con las pruebas que hicimos no funciono con la actualización que estaba en el laboratorio de redes, es decir con eclipse helios, el proyecto no reconoce una librería.
2. **Win TK:** Este es para el que funcione el sistema operativo.
3. **JPK.jar**
4. **JDK para 32 bits.**
5. **JPK.DLL.**

CLASES (**ver anexo diagrama de clases**):

NEGOCIO:

ARP:

Esta clase se encarga de toda la comunicación de la red tanto adquirir la ip, hasta conseguir el tamaño de la máscara, adquirir la máscara de red y de mandar el mensaje ARP a una máquina. Uno de sus métodos principales son los siguientes:

1. **sendARPTo:** Dado un objeto de clase device este método le manda un mensaje ARP a este. en caso de que sea exitoso el envío muestra un mensaje que la máquina está activa y retorna verdadero. De lo contrario y si hay algún error muestra alguna excepción y retorna falso.
2. **getNetworkInterfaceFor:** Este método utiliza un arreglo de tipo NetworkInterface donde verifica que la ip recibida por parámetro corresponda a una red local en caso de que sea verdadero retorna el device en caso contrario retorna null.
3. **Arp:** este método recibe como atributos la ip, la mac y un dispositivo y con esta información crea y manda el mensaje ARP, y verifica que se haya mandado el mensaje.

Device:

Esta clase controla y contiene todos los atributos de una máquina. Como la mac, ip, el tipo de dispositivo que es, el estado en el que está el dispositivo, el total de los mensajes enviados, los mensajes fallidos y los mensajes que respondidos. Este tiene los gets y los sets de todos sus atributos. Como última función tiene la suma y controla el total de los mensajes mostrados por la interfaz.

DeviceManger: Esta función tiene un mapa del total de los dispositivos, sus métodos son para agregar un dispositivo, poner el modo automático del sistema, guardar los dispositivos y lo más importante es cuando llega un nuevo dispositivo empieza el hilo para su ejecución y el conteo de los mensajes

INTERFACE:

Esta parte controla toda la interfaz del proyecto y se divide por varias clases ControlPanel esta clase se encarga de la interactividad de los botones con el usuario (**ver paso 2**), el device panel se encarga de mostrar los dispositivos que están en la red, con su imagen respectiva. El console Panel muestra los procesos que sigue el sistema para agregar el dispositivo, es decir si encontró el dispositivo y el estado del mensaje, por otro lado indica si en modo automático y manual si está mandando mensajes.

RESOURCES:

Esta carpeta se utiliza para guardar todas las imágenes utilizadas en el proyecto para indicar de qué tipo de dispositivo se encontraba.

UTILITIES:

Esta carpeta solo tiene una clase llamada utils, esta se encarga de hacer métodos que utilicen todas las clases como convertir de un arreglo de bytes a string o convertir un atributo de gregorian calendar a un string.