Escuela Politécnica Superior de Elche

SISTEMAS INFORMÁTICOS EN TIEMPO REAL
2º Ingeniería Industrial

PRÁCTICAS DE TCP-IP
ARP/ICMP/IP

Luis Miguel Jiménez

Departamento de Ingeniería de Sistemas Industriales
Área de Ingeniería de Sistemas y Automática

1. OBJETIVO

Las prácticas de TCP-IP están enmarcadas dentro del contexto de la docencia de la asignatura de "*Sistemas informáticos de Tiempo Real*" correspondiente a la titulación de Ingeniería Industrial.

Estas prácticas se proponen como una herramienta fundamental para asimilar los conceptos de sistemas distribuidos, protocolos de comunicaciones y programación de procesos distribuidos en una red de computadores, centrándose en el estudio de los protocolos de comunicaciones de la familia TCP/IP. Se estudiarán los siguientes protocolos:

- Protocolo ARP
- Protocolo ICPM
- Protocolo IP
- Protocolo TCP

2. MATERIAL EMPLEADO

Las primeras prácticas se realizan de forma individual. La segunda parte de estas prácticas requerirán el uso de dos PC (cliente y servidor) por lo que deberán realizarse en grupos de dos personas. Los equipos disponen de S.O. Windows XP. Para realizar las prácticas, el alumno debe disponer también de los apuntes de la asignatura.

El PC actuará como monitor de red mediante el programa **WireShark**, se trata de un programa gratuito para monitorizar redes. Está instalado en los PCs de laboratorio y puede descargarse del la página web de la asignatura.

El mismo PC, será utilizado como máquina generadora de mensajes de red. Para ello se utilizara una ventana de comandos en línea (**Menú Inicio -> ejecutar -> cmd**).

En la segunda parte de estas prácticas utilizaremos el programa **Socket Workbench** que nos permite establecer conexiones TCP, tanto en modo cliente como en modo servidor, utilizando la librería de sockets. (en la página web de la asignatura está disponible una versión de evaluación de este programa).

Antes de comenzar las prácticas debe leerse detalladamente la descripción de las aplicaciones utilizadas (**apartado 3**)

3. PROGRAMA WIRESHARK

El programa **Wireshark** (antiguamente llamado **Ethereal**) permite monitorizar los mensajes presentes en la red pudiendo establecer filtros para las diferentes capas, protocolos y direcciones. La figura 1 muestra la ventana principal del programa al iniciarlo.

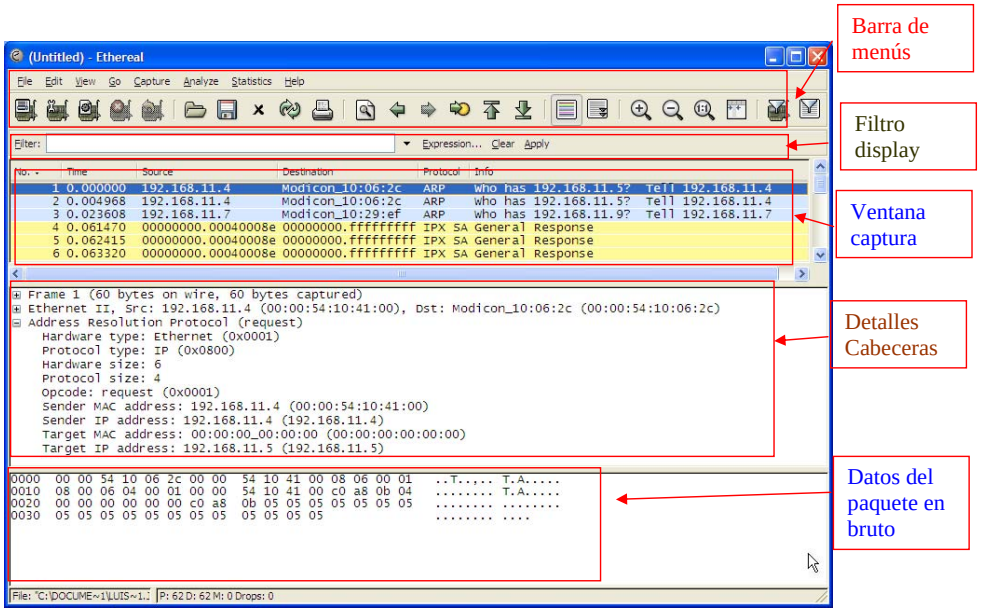


Figura 1 Ventana Principal del programa WireShark

La ventana principal del programa se divide en varias zonas:

1. **Zona superior**: muestra los menús de acceso a las opciones del programa junto con una barra de botones con los accesos directos a las opciones más usuales.
2. **Barra de Filtros**: nos permite establecer filtros de visualización de las tramas capturadas.
3. **Ventana de captura**: muestra un resumen de los paquetes capturados, indicando: número de orden, tiempo de captura, direcciones fuente y destino (MAC/IP), protocolo e información resumida de la cabecera.
4. **Ventana detalle de cabeceras**: para el paquete seleccionado en la ventana anterior se muestran debajo todos los detalles de las diferentes cabeceras que contiene. Esta es la información más importante que utilizaremos para estudiar los protocolos.
5. **Ventana de Datos**: Por último en la zona inferior se muestra el contenido en bruto del paquete seleccionado incluyendo las cabeceras y los datos transportados (se muestra en hexadecimal a la izquierda y en código ASCII a la derecha)

La información que circula por un segmento de red es ingente, por lo que verdadera potencia del programa es la capacidad para especificar filtros que permiten seleccionar la información buscada (protocolos, direcciones origen o destino,…). El programa ***WireShark*** permite establecer dos tipos de filtros:

1. **Filtros de captura**: El programa solo almacenará en memoria los paquetes que cumpla la restricciones que especifiquemos. Este es el tipo de filtros que utilizaremos en estas prácticas. La sintaxis utilizada es compatible con la utilidad de UNIX **tcpdump**. En la página web de la asignatura está disponible un tutorial detallado de esta sintaxis. El programa permite guardar los filtros en disco para su uso posterior, disponiendo así mismo de una decena de filtros comunes ya predefinidos.

2. **Filtros de Visualización**: para todos los paquetes almacenados en memoria (capturados) permite mostrar aquellos que cumplen una condición fijada por el usuario. La sintaxis utilizada es específica de WireShark por lo que se recomienda leer la ayuda del programa. Se dispone de un asistente (botón ***Expression***) que facilita esta tarea.

Para realizar una captura de datos por primera vez pulsaremos ***Capture/Options*** en el menú, o bien el botón  . Nos mostrará una ventana (Figura 2) donde configuraremos las opciones de captura.

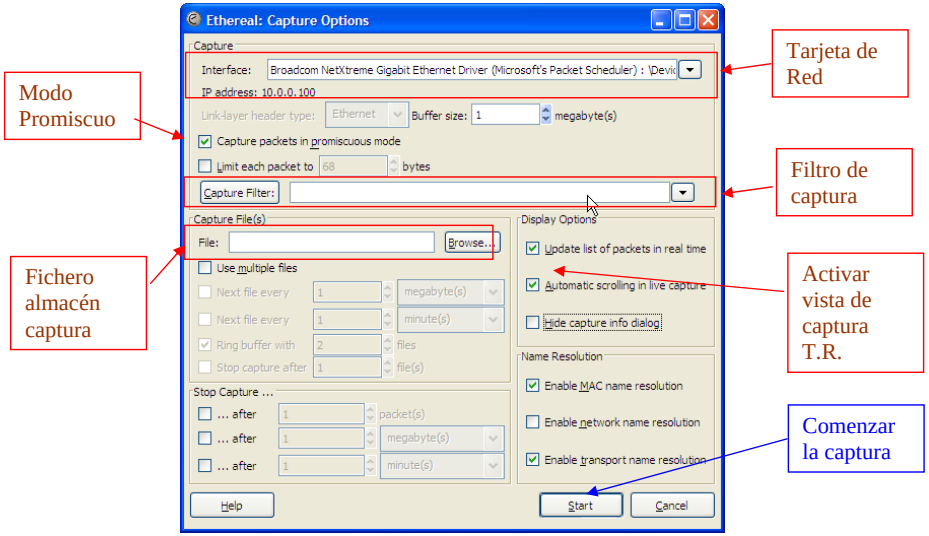


Figura 2 Ventana Opciones de Captura

Seleccionaremos la tarjeta de red, si disponemos de varias, activaremos el modo promiscuo (captura todos los paquetes) e indicaremos el filtro de captura. Si deseamos guardar los paquetes capturados indicaremos el nombre del fichero, en caso contrario se almacenan en memoria perdiéndose al terminar o realizar una nueva captura (el programa nos preguntará siempre si deseamos guardar las tramas antes de borrarlas). Activaremos así mismo la visualización en tiempo real de los paquetes capturados tal como se indica en la figura 2.

Una vez fijada la configuración pulsaremos el botón ***START*** para empezar la captura. Nos mostrará una nueva ventana (Figura 3) con el progreso de la captura. Muestra en número de paquetes capturados, clasificados según el protocolo. Cuando deseemos terminar pulsaremos el botón ***STOP***. Volveremos de este modo a la ventana principal (figura 1) mostrando los paquetes capturados.

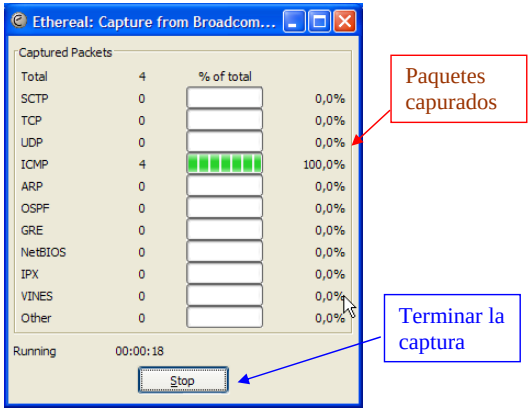


Figura 3 Ventana de captura en curso

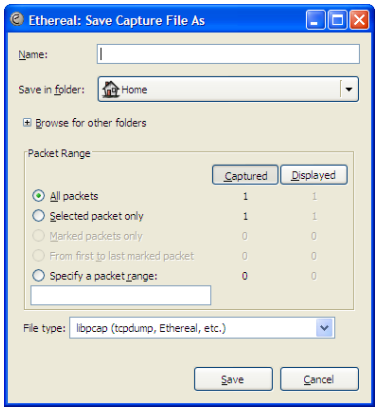


Figura 4 Ventana para guardar resultados

En capturas sucesivas el programa nos solicitará si deseamos almacenar los datos anteriores. En caso afirmativo indicaremos el fichero y la carpeta donde se almacenarán las tramas (Figura 4).

LISTA DE COMANDOS DE RED Windows /MacOSX/UNIX

Para realizar la práctica deberemos utilizar también algunos comandos de red por lo que se resumen a continuación con su sintaxis:

```
winipcfg   (Windows 95)             Muestra configuración IP/MAC

ipconfig /all  (Windows NT/XP)      Muestra la configuración IP/MAC
ipconfig /renew  (Windows NT/XP)    Renueva la dirección IP
ipconfig /release  (Windows NT/XP)  Libera la dirección IP actual

/sbin/ifconfig -a (MacOSX/UNIX)          Muestra la configuración IP/MAC

ping: Envía tramas ICMP (solicitud de eco). Opciones Windows:
            <ipaddr>    Dirección ip destino
            -n <x>      Envía <x> paquetes
            -l  <x>     Envía paquetes de longitud <x>
            -f          Activa el bit 'Don't fragment'
            -i <x>      Limita la vida del paquete (TTL) a <x> saltos
            -r <x>      Graba la ruta del eco para <x> saltos
            -s <x>      Graba la los tiempos de paso para <x> saltos
            -w <x>      Tiempo de espera del eco en milisegundos <x>

arp -a                  Muestra la tabla ARP
arp -d <ipaddr>         Borra la entrada <ipaddr> de la tabla arp

netstat -rn:                        Visualiza las rutas existentes
route delete <ipaddr>               Borra la ruta hacia <ipaddr>
route add <ipdst> <mask> <gateway>  Añade una ruta manual

tracert  <ipdst>              Obtiene la ruta hasta la ip/nombre destino
```

4. TAREAS

Para la realización de las prácticas utilizaremos la red de la Universidad. Se trata de una red privada de clase A (**10.x.x.x**) estructurada en subredes. (El guión de la prácticas esta diseñado para que pueda realizarse en cualquier red sustituyendo las etiquetas que indicaremos a continuación).

Al iniciar la práctica deberemos consultar los siguientes datos de configuración de nuestra conexión de red. Para ello ejecutaremos el siguiente comando en una consola:

Windows NT/2000/XP/Vista: **ipconfig /all**
Windows 95/98**:** **winipcfg**
Linux/MacOSX: **/sbin/ifconfig –a**

A partir de los datos mostrados apuntaremos lo siguiente:

- **Dirección MAC/Ethernet** <**MAC_HOST**>**:** (dirección física)
- **Dirección IP/inet** <**IP_HOST**>:
- **Máscara de Subred** <**MASK**>
- **Pasarela por defecto:** <**GATEWAY**>**:** (puerta de enlace predeterminada)

Nota importante: el resto del guión se basará en el valor obtenido para estos tres datos. Deberemos sustituir la etiqueta correspondiente por el valor asignado a nuestro equipo al inicio de cada sesión (La **IP_HOST** y el **GATEWAY** pueden cambiar)

La notación a emplear para la dirección IP será la decimal puntuada de cuatro campos:
(ejemplo: **10.1.40.32**)

La notación usada para una dirección MAC será la hexadecimal de seis campos separados por el símbolo'**:**', (ejemplo: **00:E0:4C:E8:99:9F**)

Aquellos apartados que requieran dos equipos será necesario disponer de los datos correspondientes a cada uno.

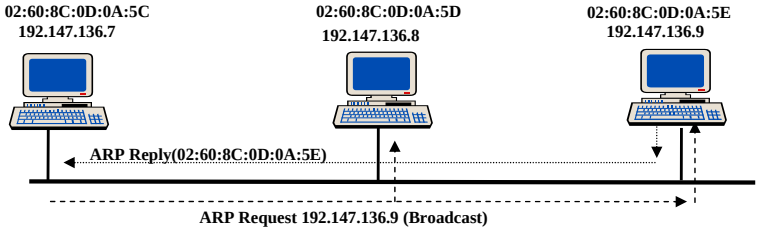
En cada ejercicio deberá guardarse las tramas capturadas con el programa WireShark en un fichero para realizar posteriormente los informes de prácticas.

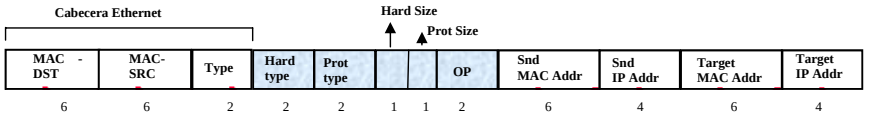
Cuestión 1. Protocolo ARP

Funcionamiento general y formato de trama ARP:

El protocolo ARP permite averiguar la dirección MAC correspondiente a una dirección IP. El protocolo actúa de la siguiente forma:

- La máquina origen envía una trama de '*Petición ARP'* a todas las máquinas de la red (*broadcast*)
- El protocolo ARP de la máquina destino verifica que el solicitante pide su dirección IP.
- La máquina de destino emite una trama de *'Respuesta ARP'* incorporando su dirección MAC e IP
- La máquina origen recibe la respuesta guardando el resultado en una tabla, pudiendo emitir a partir de entonces datagramas IP





Type: 0x0806 (trama ARP)
Hard type: 1 (Ethernet)
Prot. type: 0x0800 (IP)
Hard size: 6 (bytes dir. MAC)
Prot size: 4 (bytes dir. IP)

OP: tipo de operación
1: ARP request
2: ARP reply
3: RARP request
4: RARP reply

Tareas a realizar:

1. Arrancar el programa ***WireShark*** y abrir la ventana de configuración de captura. Realizaremos la configuración descrita en el apartado 3.

2. Para discriminar tramas **ARP** utilizaremos el siguiente filtro de captura:

   Filtro Captura=> **arp and ether host <MAC_HOST>**

*Debemos cambiar la etiqueta <**MAC_HOST**> por la dirección MAC apuntada anteriormente (utilizaremos el símbolo **':'** para separar cada byte expresado en **hexadecimal)***

3. Iniciaremos la captura pulsando **START**.

4. Abriremos una consola y comprobaremos que la caché ARP no contiene la dirección IP de destino utilizada, si ya existiese en la tabla lo borraremos con el comando **arp –d**:

```
C:\>arp -a
C:\>arp -d  <GATEWAY>
```

Nota: *la dirección IP indicada* **<GATEWAY>** *corresponde la pasarela por defecto que* ***deberemos sustituir*** *por la obtenida al principio de la práctica.*

5. En la misma consola lanzaremos peticiones *'echo'* a través del programa **ping:** Petición de eco con un solo paquete

```
C:\>ping -n 1  <GATEWAY>
```

6. Cuando se visualicen en la ventana las dos tramas pararemos la captura pulsando el botón **STOP**. Es recomendable **guardar en un fichero** las tramas capturadas.

7. Verificar de nuevo el contenido de la caché **ARP** ejecutando el comando **arp**. Anotar el resultado.

```
C:\>arp -a
```

8. Estudiar detenidamente el contenido de las tramas presentes para responder a las siguientes preguntas:

Preguntas:

a) Descríbase la secuencia de tramas involucradas, justificando todas las direcciones MAC e IP que aparecen. Representarlo gráficamente
b) ¿Cuál es el estado de la memoria caché de **ARP** después del **ping**?

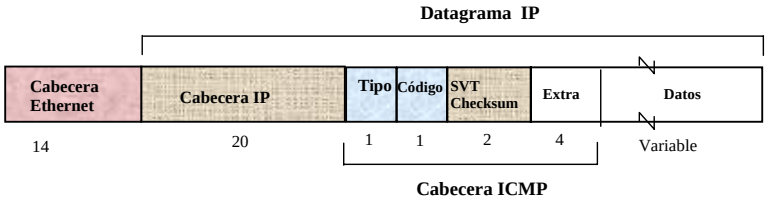
Cuestión 2. Protocolo ICMP/Ping

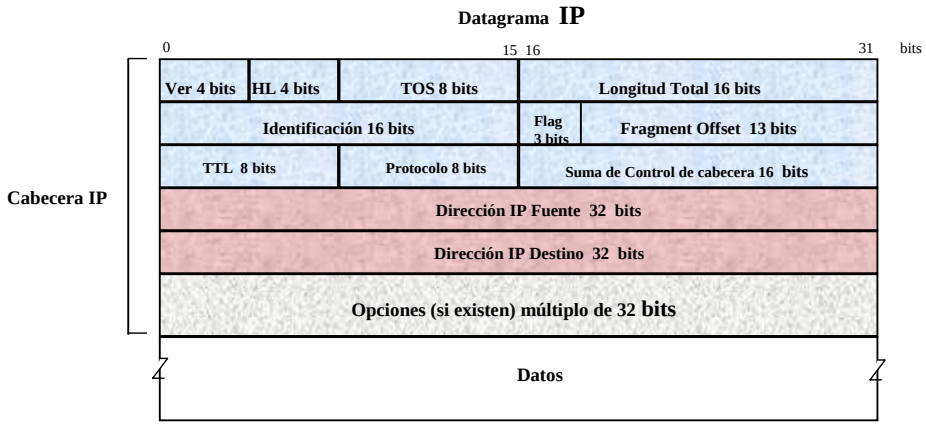
Descripción del Protocolo PING:

Realiza una petición de eco sobre una máquina remota, permitiendo comprobar errores de comunicación, rutas y el rendimiento de la red.

Ping utiliza el comando eco de **ICMP** para enviar un datagrama a su destinatario y esperar su retorno. De este modo es capaz de evaluar tiempos de respuesta promedios. Al concluir el comando, queda reflejado el número de paquetes perdidos, los tiempos mínimos, máximos y medios de respuesta (ida y vuelta). Nos permitirá conocer la tasa de error de un enlace así como la velocidad real de transmisión de forma experimental.

Dispone de varias opciones, entre las que cabe destacar la posibilidad de modificar el tamaño del paquete enviado, el registro de ruta, y el control del número de paquetes enviados (ver página 3). El formato de datagrama corresponde a un datagrama **ICMP** que incluye junto a los datos, 20 bytes de la cabecera **IP** y 8 bytes de la **ICMP**. Adicionalmente está precedido de la cabecera **Ethernet** (14 bytes). donde se indica en tiempo de respuesta del eco **ICMP** y el número de secuencia.





Tareas a realizar:

1. En el Programa **WireShark** iniciar una nueva captura con el siguiente filtro:

   Filtro Captura=> **icmp and ip host <IP_HOST>**

Debemos cambiar la etiqueta <**IP_HOST**> por la dirección IP apuntada anteriormente (utilizaremos el símbolo **'.'** para separar cada byte expresado en **decimal**

2. En una consola ejecutar :

```
C:\>ping –n 1  193.147.145.166
```

3. Parar la captura y estudiar las tramas involucradas. Es recomendable **guardar en un fichero** las tramas capturadas.

Preguntas:

a) ¿Qué tipos de mensajes ICMP aparecen?
b) Estudiar y describir el contenido de las cabeceras IP/ICMP.
c) Justificar la longitud de los paquetes de acuerdo a los formatos de trama de cada protocolo involucrado. ¿Cual es el contenido del campo **extra de la cabecera ICMP** asociado a la petición ping (*echo request*)?

Cuestión 3. Fragmentación IP

Manteniendo los mismos filtros de la cuestión anterior en el programa monitor **WireShark** (en modo captura), ejecutaremos:

```
C:\>ping –n 1  -l 2000  193.147.145.166
```

a) Descríbase el número de fragmentos correspondientes a cada datagrama
b) Descríbase y justifíquese la información presente en los campos ***flags***, ***identificador*** y ***fragoff*** de la cabecera IP en cada datagrama.
c) ¿Cuánta información extra es enviada, debida a la fragmentación, por cada datagrama original ? (Nota: calcular y justificar la diferencia entre los bytes transferidos y los que se transferirían si no hubiera fragmentación)
d) Determínese el MTU de la máquina emisora.

Cuestión 4. ICMP TTL_exceeded

Manteniendo los mismos filtros de la cuestión anterior en el programa monitor **WireShark** (en modo captura), ejecutaremos:

```
C:\>ping –n 1  -i 1  193.147.145.166
```

Nota: el parámetro -i <x> pone el campo TTL del datagrama enviado al valor <x>

a) ¿Quién envía el mensaje ICMP TTL_exceeded ?
b) ¿Cuál fue la causa del error?
c) Jugando con el parámetro TTL del paquete, calcular el número de saltos (routers) por los que debe pasar el datagrama hasta llegar a su destino. Anotar la IP del nodo que nos devolvió el mensaje icmp.

Con el programa **WireShark** en modo captura ejecutar el siguiente comando:

```
C:\>tracert  193.147.145.166
```

d) Comparar la salida del comando con los datos obtenidos en el apartado anterior. Estudiar los mensajes enviados por **tracert** para evaluar la ruta.

e) Calcular la ruta a la siguiente dirección IP: **18.7.22.69**
¿Dónde está ubicado dicho servidor?