3.2.1 IP/ICMP

En esta sesión investigaremos el protocolo IP. Analizaremos paquetes IP enviados y recibidos y su fragmentación. Entenderemos cómo funciona el protcolo ICMP y algunas de sus aplicaciones mediante los procesos *ping* y *traceroute*.

3.2.1.1 ICMP

ICMP (Internet Control Message Protocol) es un subprotocolo de control y notificación de errores de IP. Su funcionalidad básica es la de comunicar errores en la red, como el de no ser posible llegar a un nodo y host determinado.

En principio, ICMP no es usado por una aplicación de nivel superior (como ocurre con TCP y UDP). No obstante existen un par de procesos que están desarrollados sobre ICMP y que nos servirán para ver los pasos que se llevan a cabo al enrutar sobre IP.

El formato del mensaje ICMP es el siguiente:

A modo de ejemplo, haremos uso de los mensajes Echo Request/Reply, que corresponden a los tipos 0 y 8 respectivamente y código 0.

Más información acerca del protocolo ICMP en http://www.rfc-es.org/rfc/rfc0792-es.txt.

3.2.1.2 **PING**

Utilizaremos el proceso ping para estudiar los mensajes ICMP.

La aplicacion ping comprueba el estado de la conexión con uno o varios equipos remotos por medio de los paquetes de solicitud de eco y de respuesta de eco (ambos definidos en el protocolo de red ICMP) para determinar si un sistema IP específico es accesible en una red. Es útil para diagnosticar los errores en redes o enrutadores IP.

1. Abrir una consola de MS-DOS.
2. Abrir Wireshark y comenzar a capturar filtrando por ICMP.
3. En la consola de MS-DOS, introducir el comando "*ping –n 2 ip_destino*" sin comillas y siendo *ip_destino* la IP de algún equipo de la sala.
4. Detener la captura de Wireshark.

La captura debería mostrar algo parecido a lo siguiente:

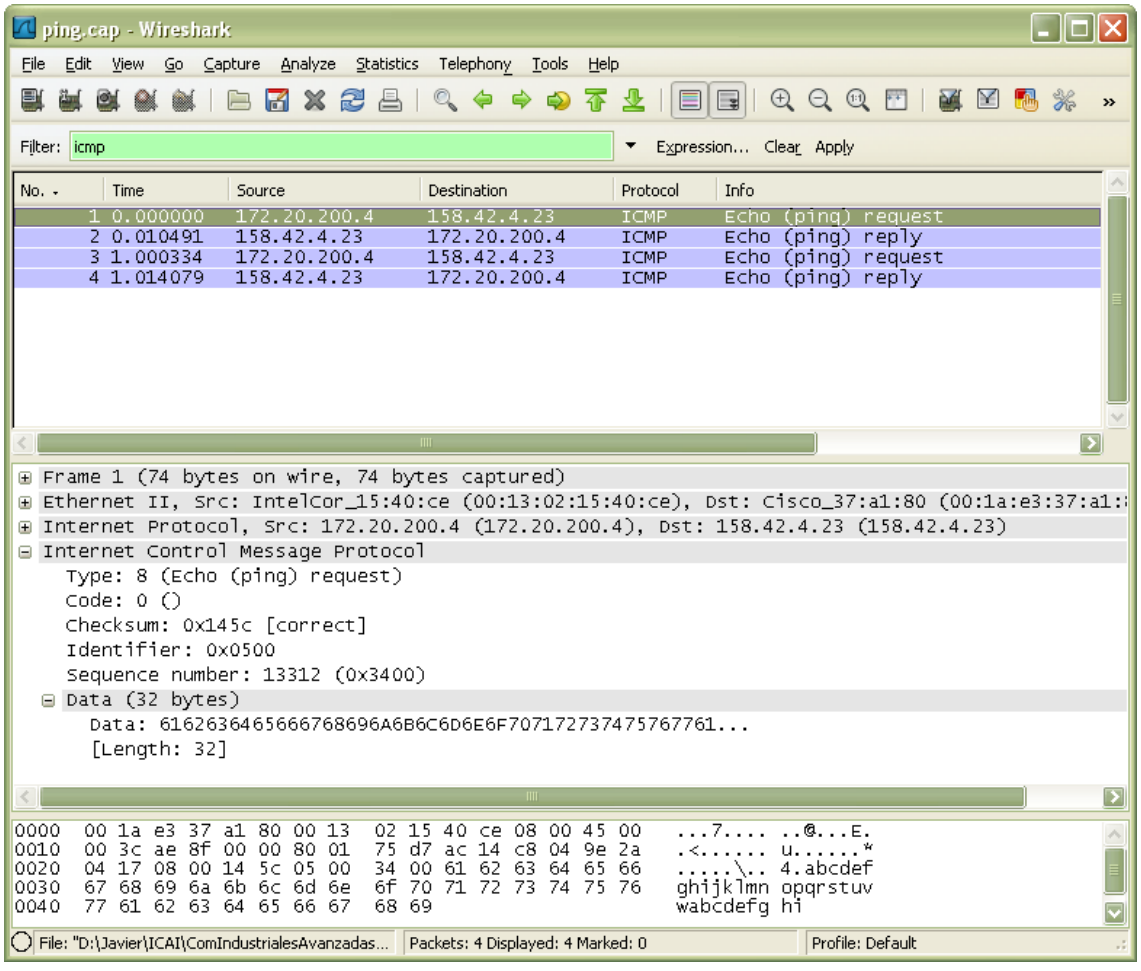


Imagen 1 - Captura comando ping Wireshark

1. *Explica la opción utilizada en el comando ping. Busca en la ayuda (ping –h).*
2. *¿Cuántos mensajes ICMP se han intercambiado? ¿Por qué?*
3. *Copia los valores de los campos "Type", "Code", "Identifier" y "Sequence Number" del primer par de mensajes Echo request/reply y explica los valores.*

3.2.1.3 **TraceRoute**

Otro proceso que hace uso del protocolo ICMP es el traceroute (tracert para window). Este programa envía en primer lugar uno (o varios) mensajes ICMP (Echo Request) encapsulados en un paquete IP con TTL=1 (Time To Live). Tras recibir la correspondiente respuesta, envía otro mensaje ICMP (Echo Request) con TTL=2. Continúa así sucesivamente.

Cada vez que un paquete alcanza un router, éste chequea el valor del TTL y le descuenta uno. En el caso de que el valor del TTL sea 0, el router contesta al originante del mensaje con un mensaje ICMP (tipo 11 – TTL-exceeded).

Como resultado de este comportamiento, un paquete con TTL=1 dará lugar a que el primer router que se encuentre le mensaje ICMP lo devuelva al originante. El paquete con TTL=2 dará lugar a que el segundo router por el que pase el mensaje lo de vuelva por la razón explicada anteriormente y así sucesivamente.

Analicémoslo con Wireshark:

1. Abrir Wireshark
2. Abrir una consola de MS-DOS e introducir el comando "tracert 192.168.56.200".
3. Detener la captura de Wireshark una vez traceroute haya finalizado.

En la consola de MS-DOS podremos ver los saltos que se van realizando entre los nodos de la red. En este caso únicamente tenemos un router entre cada equipo del laboratorio y 192.168.56.200.

Observad cómo en la captura de Wireshark en valor del TTL en los diferentes mensajes ICMP se va incrementando.

3.2.1.4 **Fragmentación IP**

A continuación analizaremos la fragmentación sobre IP.

1. Abrir una consola de MS-DOS.
2. Abrir Wireshark y comenzar a capturar filtrando por IP.
3. En la consola de MS-DOS, introducir el comando "*ping –l 3000 ip_destino*" sin comillas y siendo *ip_destino* la IP de algún equipo de la sala.
4. Detener la captura de Wireshark.

Tras la captura de Wireshark podemos ver cómo se forman los mensajes ICMP a base de paquetes IP.

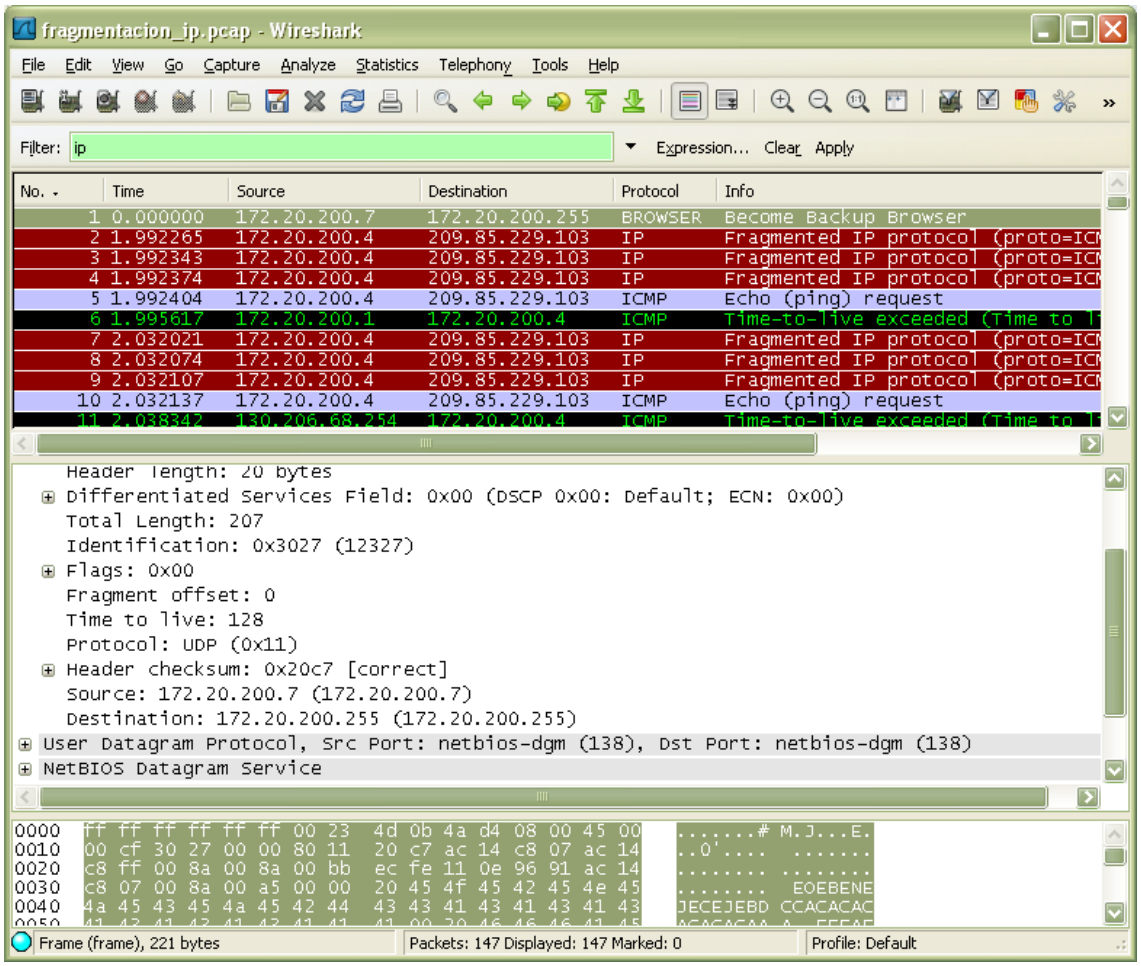


Imagen 2 - Captura Wireshark fragmentación IP

4. *Describir la opción usada en el comando ping.*
5. *¿Qué información en la cabecera de IP indica que se trata de un paquete fragmentado?¿Cómo puedes identificar el paquete con el primer fragmento?*
6. *¿Qué tamaño tiene el paquete IP?*

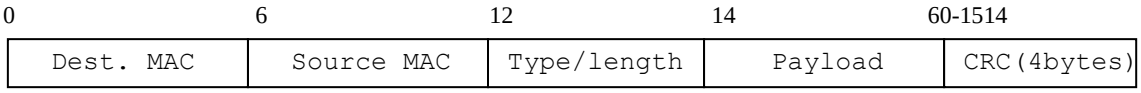
7. *¿Cómo podemos saber si aún quedan más paquetes fragmentos por venir?*
8. *¿Cómo identificaremos el paquete portador del último fragmento?*
9. *¿Cuánta información extra es enviada, debida a la fragmentación, por cada paquete original?(Se pide calcular y justificar la diferencia entre los bytes transferidos y los que se transferirían si no hubiera fragmentación.)*

3.2.2 Ethernet

En esta sesión, investigaremos el protocolo Ethernet y ARP.

3.2.2.1 Captura de trazas Ethernet

El formato de una traza Ethernet es el siguiente:



Para verlo de forma práctica con Wireshark, hagamos lo siguiente:

1. Abrir un navegador web y limpiar los archivos de caché.
2. Abrir Wireshark y comenzar a capturar filtrando por http.
3. Acceder a la dirección : 192.168.56.200
4. Parar la captura de Wireshark.

Ya que esta práctica trata sobre Ethernet y ARP, no estamos interesados en IP o en capas de protocolos de más alto nivel. De manera que, para contestar a las siguientes cuestiones hemos de fijarnos en la traza Ethernet que encapsula el mensaje GET HTTP.

10. *¿Cuál es la dirección Ethernet de vuestro equipo?*
11. *¿Cuál es la dirección Ethernet destino de la trama?¿Es la misma que la dirección Ethernet de 192.168.56.200? (Pista: la respuesta es no) ¿Quién tiene esa dirección?*
12. *¿Qué indican los bytes correspondientes al campo "Type"?¿Qué valor tienen?*

3.2.2.2 **ARP INTRANET**

Veamos ahora como actúa el protocolo ARP (Arddress Resolution Protocol) sobre Ethernet.

Es un protocolo de nivel de red responsable de encontrar la dirección hardware (Ethernet MAC) que corresponde a una determinada dirección IP. Para ello se envía un paquete (ARP request) a la dirección de difusión de la red (broadcast (MAC = xx xx xx xx xx xx)) que contiene la dirección IP por la que se pregunta, y se espera a que esa máquina (u otra) responda (ARP reply) con la dirección Ethernet que le corresponde. Cada máquina mantiene una caché con las direcciones traducidas para reducir el retardo y la carga.

Hagamos lo siguiente:

1. Limpiad la caché ARP del equipo (consultad la ayuda tecleando el comando arp en una consola de MS-DOS).
2. Limpiad la caché del navegador web que vayáis a utilizar.
3. Abrid Wireshark y comenzar a capturar filtrando por HTTP.
4. Acceded a la página web del servidor web que se encuentre en vuestra subred.
5. Parad la captura de Wireshark, limpiar el campo de filtrado.

La captura mostrará algo parecido a la Imagen 3.

Observamos como, antes de lanzar la traza Ethernet encapsulando el mensaje HTTP (sobre TCP/IP), existen dos mensajes ARP. Analicémoslos con las siguientes cuestiones:

13. *¿Cuáles son los valores hexadecimales de las direcciones origen y destino del mensaje ARP?*
14. *¿Cómo podemos saber que la trama Ethernet encapsula un mensaje ARP?*
15. *¿Cuál es la dirección Ethernet destino en el mensaje ARP-request?¿Por qué tiene ese valor?*
16. *¿A qué corresponden los valores hexadecimales de las direcciones Ethernet en el mensaje ARP-reply?*

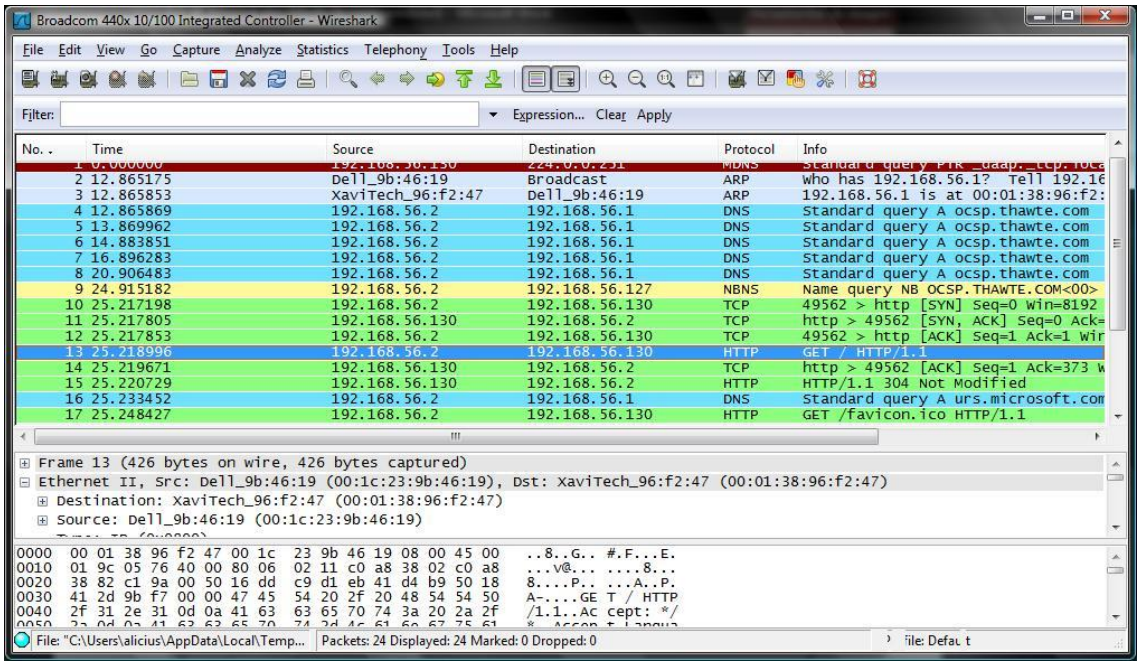


Imagen 3 - Captura ARP con Wireshark