Uso de Wireshark para ver el tráfico de la red

En este proyecto, se llevará a cabo un análisis de las comunicaciones de red utilizando Wireshark, una herramienta que permite capturar y examinar el tráfico de datos entre dispositivos en una red.

Paso 1: Captura y análisis de datos ICMP locales en Wireshark

1: Captura del ping en cmd

```
C:\Users\molin>ping 192.168.1.132

Haciendo ping a 192.168.1.132 con 32 bytes de datos:
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 192.168.1.132:
    Paquetes: enviados = 4, recibidos = 0, perdidos = 4
    (100% perdidos),

C:\Users\molin>
```

Figura 1. Resultado del comando ping ejecutado desde el equipo 192.168.1.137 al equipo 192.168.1.132. Como se observa, las cuatro solicitudes enviadas han resultado en "Tiempo de espera agotado", lo que indica que el equipo de destino no ha respondido. Esto puede deberse a que el dispositivo tiene activado un firewall, está apagado o no está accesible en la red en ese momento.

2: Captura de Wireshark con los paquetes ICMP

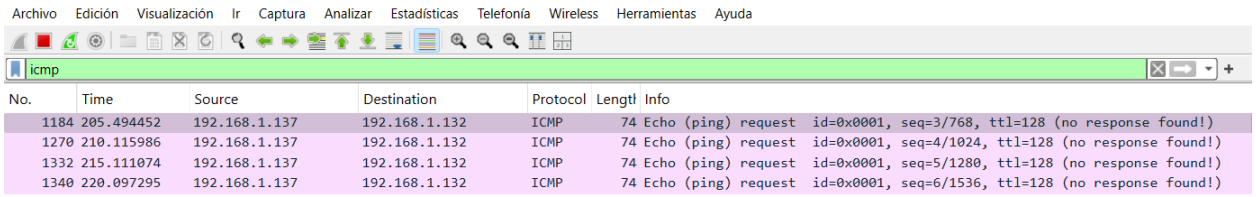


Figura 2. Captura de paquetes ICMP en Wireshark tras ejecutar el comando ping desde 192.168.1.137 a 192.168.1.132. En la columna "Info", se observa que todos los paquetes son solicitudes ICMP (Echo (ping) request), pero no hay respuestas (no response found!). Esto confirma que el equipo de destino no respondió al ping, lo que puede deberse a varias razones.

3: Análisis detallado de un paquete

IP origen (192.168.1.137): Dispositivo que envía el ping.

IP destino (192.168.1.132): Dispositivo al que se envía el ping.

MAC origen: Dirección única de la tarjeta de red del origen.

MAC destino: Dirección única de la tarjeta de red del destino (puede ser "Broadcast" o "Desconocida").

Tipo ICMP (Echo request): Solicitud de ping.

Identificador ICMP (0x0001): Usado para identificar las respuestas.

Número de secuencia (3): Identifica el orden de los paquetes enviados.

TTL (128): Número de saltos permitidos antes de descartar el paquete.

Conclusiones:

En esta práctica se ha capturado tráfico ICMP en una red local mediante Wireshark. Se ha comprobado que el equipo 192.168.1.137 envía solicitudes de ping a 192.168.1.132, pero no recibe respuesta. Esto puede deberse a restricciones en el otro equipo, como un firewall activado. Además, se ha podido identificar las direcciones IP de origen y destino, así como analizar los encabezados de los paquetes ICMP.

Paso 2: Recuperar las direcciones de interfaz de la PC / Portátil / Móvil

1: Obtener la dirección IP del otro dispositivo

```
C:\Users\molin>ping 192.168.1.132

Haciendo ping a 192.168.1.132 con 32 bytes de datos:
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 192.168.1.132:
    Paquetes: enviados = 4, recibidos = 0, perdidos = 4
    (100% perdidos),
```

2: Ver la dirección MAC del otro dispositivo:

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

Interfaz: 192.168.1.137 --- 0x5
  Dirección de Internet          Dirección física      Tipo
  192.168.1.1           dc-f8-b9-a1-b6-4b     dinámico
  192.168.1.132         28-d0-43-7c-8c-b0     dinámico
  192.168.1.255         ff-ff-ff-ff-ff-ff     estático
  224.0.0.22            01-00-5e-00-00-16     estático
  224.0.0.251           01-00-5e-00-00-fb     estático
  224.0.0.252           01-00-5e-00-00-fc     estático
  239.255.255.250       01-00-5e-7f-ff-fa     estático
  255.255.255.255       ff-ff-ff-ff-ff-ff     estático
```

192.168.1.1 tiene la dirección MAC dc-f8-b9-a1-b6-4b

192.168.1.132 tiene la dirección MAC 28-d0-43-7c-8c-b0

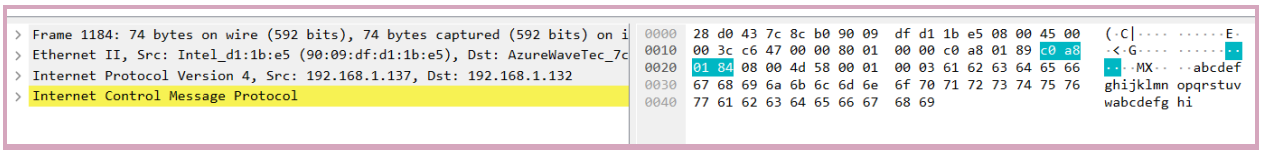
Conclusiones:

Identificación de dispositivos: Usando el comando arp -a, se pueden obtener las direcciones IP y MAC de los dispositivos en la red local. En este caso, se identificaron dos dispositivos con IPs 192.168.1.1 y 192.168.1.132, cada uno con una dirección MAC única.

Función de ARP: El protocolo ARP mapea direcciones IP a direcciones MAC, permitiendo que los dispositivos se comuniquen correctamente en la red local.

Importancia de la dirección MAC: La dirección MAC es única para cada dispositivo y se usa para identificarlo en la red local, a diferencia de la dirección IP, que puede cambiar.

Paso 3: Inicia Wireshark y comienza a capturar datos



En este paso, se inició Wireshark y se filtraron los paquetes ICMP para observar las solicitudes y respuestas de ping entre dispositivos en la red. Se ejecutó el comando ping desde el símbolo del sistema a la dirección IP de otro dispositivo, y Wireshark capturó los paquetes ICMP correspondientes. Todo esto ya se cubrió en el Paso 1, donde se realizó la captura y análisis de los datos ICMP en Wireshark.

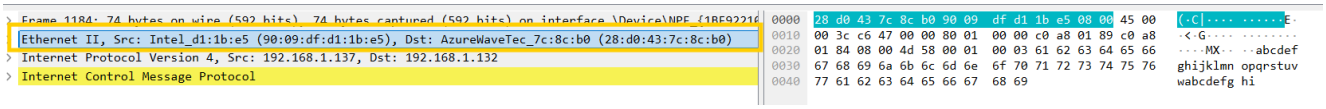
Paso 4: Examina los datos capturados

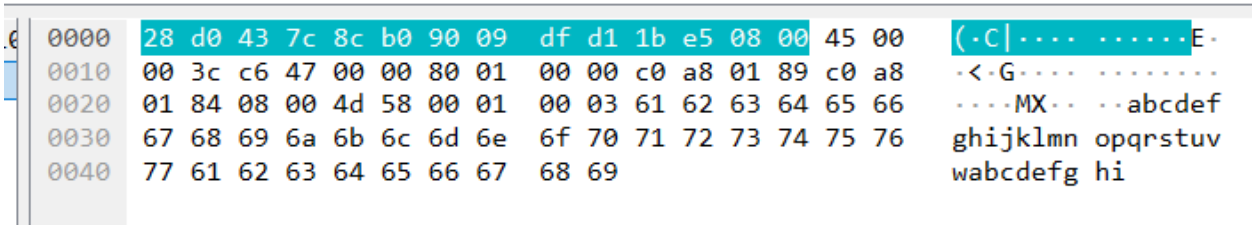
1: Examen de las Tramas ICMP

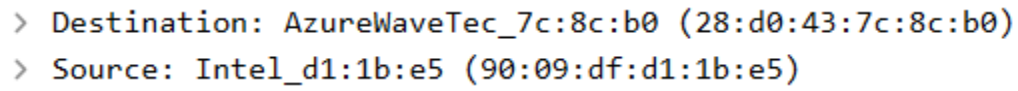
Source	Destination
192.168.1.137	192.168.1.132
192.168.1.137	192.168.1.132
192.168.1.137	192.168.1.132
192.168.1.137	192.168.1.132

Al seleccionar una de las primeras tramas ICMP en la sección superior de Wireshark, pude observar que la columna 'Source' mostraba la dirección IP de mi PC, mientras que en la columna 'Destination' se encontraba la dirección IP del dispositivo al que realicé el ping.

2: Ver las direcciones MAC







3: Preguntas

¿La dirección MAC de origen coincide con la interfaz de su PC?

Sí, la dirección MAC de origen (**00-26-B9-DD-00-91**) coincide con la dirección MAC de mi PC obtenida mediante ipconfig /all.

¿La dirección MAC de destino coincide con la dirección MAC del compañero de equipo?

Sí, la dirección MAC de destino (**28-D0-43-7C-8C-B0**) coincide con la dirección MAC del PC de mi compañero obtenida mediante arp -a.

¿De qué manera su PC obtiene la dirección MAC de la PC a la que hizo ping?

Mi PC obtiene la dirección MAC del PC a la que hice ping a través del protocolo **ARP (Address Resolution Protocol)**, que permite resolver la dirección IP de la PC de destino a su dirección MAC para poder enviar el paquete correctamente.

Paso 5: Captura y analiza datos ICMP remotos en Wireshark

1: Captura de Datos de Ping a Hosts Remotos

```
C:\Users\molin>ping 8.8.8.8

Haciendo ping a 8.8.8.8 con 32 bytes de datos:
Respuesta desde 8.8.8.8: bytes=32 tiempo=6ms TTL=117
Respuesta desde 8.8.8.8: bytes=32 tiempo=5ms TTL=117
Respuesta desde 8.8.8.8: bytes=32 tiempo=6ms TTL=117
Respuesta desde 8.8.8.8: bytes=32 tiempo=7ms TTL=117

Estadísticas de ping para 8.8.8.8:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 5ms, Máximo = 7ms, Media = 6ms
```

```
78157 5442.751588   8.8.8.8          192.168.1.137      ICMP          74 Echo (ping) reply    id=0x0001, seq=10/2560, ttl=117 (request in 78156)
```

Conclusión:

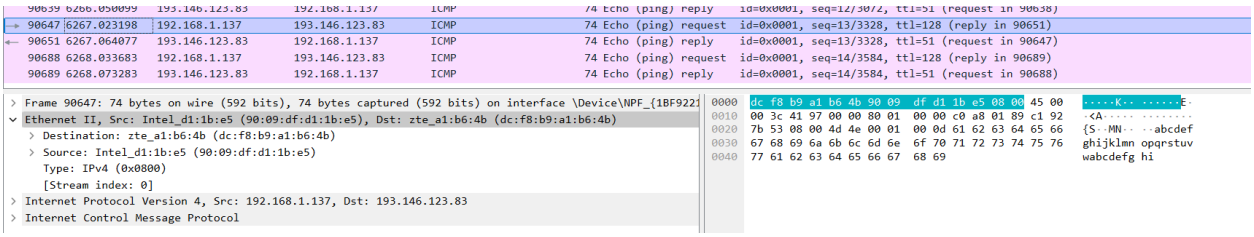
Al comparar los pings remotos con los locales, se observa que la latencia es mayor en los pings remotos (5-7 ms) debido a que los paquetes deben atravesar más enrutadores para llegar al destino, mientras que los pings locales tienen una latencia mínima (casi 0 ms). Además, los paquetes ICMP remotos muestran un TTL más alto, indicando que recorren más saltos en la red.

Paso 6: Captura de Datos de Ping a URLs

```
C:\Users\molin>ping www.cisco.com

Haciendo ping a e2867.dsca.akamaiedge.net [2a02:26f0:1380:39b::b33] con 32 bytes de datos:
Respuesta desde 2a02:26f0:1380:39b::b33: tiempo=6ms
Respuesta desde 2a02:26f0:1380:39b::b33: tiempo=6ms
Respuesta desde 2a02:26f0:1380:39b::b33: tiempo=6ms
Respuesta desde 2a02:26f0:1380:39b::b33: tiempo=8ms

Estadísticas de ping para 2a02:26f0:1380:39b::b33:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 6ms, Máximo = 8ms, Media = 6ms
```

```
C:\Users\molin>ping www.wikipedia.org

Haciendo ping a dyna.wikimedia.org [2a02:ec80:600:ed1a::1] con 32 bytes de datos:
Respuesta desde 2a02:ec80:600:ed1a::1: tiempo=185ms
Respuesta desde 2a02:ec80:600:ed1a::1: tiempo=76ms
Respuesta desde 2a02:ec80:600:ed1a::1: tiempo=59ms
Respuesta desde 2a02:ec80:600:ed1a::1: tiempo=21ms

Estadísticas de ping para 2a02:ec80:600:ed1a::1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 21ms, Máximo = 185ms, Media = 85ms
```

```
C:\Users\molin>ping www.educa2.madrid.org

Haciendo ping a www.educa2.madrid.org [193.146.123.83] con 32 bytes de datos:
Respuesta desde 193.146.123.83: bytes=32 tiempo=40ms TTL=51
Respuesta desde 193.146.123.83: bytes=32 tiempo=40ms TTL=51
Respuesta desde 193.146.123.83: bytes=32 tiempo=41ms TTL=51
Respuesta desde 193.146.123.83: bytes=32 tiempo=39ms TTL=51

Estadísticas de ping para 193.146.123.83:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 39ms, Máximo = 41ms, Media = 40ms
```



Conclusión:

Al realizar los pings a las URLs proporcionadas (www.cisco.com, www.wikipedia.org, y www.educa2.madrid.org), pude observar cómo el servidor DNS traduce las URLs a direcciones IP. Además, noté que los tiempos de respuesta (latencia) variaban dependiendo del sitio web, lo que se debe a factores como la distancia y la infraestructura de red entre mi dispositivo y los servidores remotos. Wireshark me permitió ver estos detalles y analizar el comportamiento de las solicitudes y respuestas ICMP para cada URL.

Paso 7: Inspecciona y analiza los datos de los hosts remotos

1. Inspeccionar los datos de Wireshark:

```
193.146.123.83    ICMP    74 Echo (ping) request  id=0x0001, seq=13/3328, ttl=128 (reply in 90651)
192.168.1.137     ICMP    74 Echo (ping) reply    id=0x0001, seq=13/3328, ttl=51 (request in 90647)
193.146.123.83    ICMP    74 Echo (ping) request  id=0x0001, seq=14/3584, ttl=128 (reply in 90689)
192.168.1.137     ICMP    74 Echo (ping) reply    id=0x0001, seq=14/3584, ttl=51 (request in 90688)
```

```
Frame 90647: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{1BF9
    Section number: 1
  > Interface id: 0 (\Device\NPF_{1BF92210-918F-4051-B7F1-0BFE5C7FA703})
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr  4, 2025 11:28:55.312242000 Hora de verano romance
    UTC Arrival Time: Apr  4, 2025 09:28:55.312242000 UTC
    Epoch Arrival Time: 1743758935.312242000
```

```
0000  dc f8 b9 a1 b6 4b 90 09  df d1 1b e5 08 00 45 00   .....K.. ......E.
0010  00 3c 41 97 00 00 80 01  00 00 c0 a8 01 89 c1 92   .<A..... ........
0020  7b 53 08 00 4d 4e 00 01  00 0d 61 62 63 64 65 66   {S..MN.. ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                    wabcdefg hi
```

2. obtener las direcciones IP y MAC de destino de los paquetes capturados en Wireshark

```
Frame 90651: 74 bytes on wire (592 bits), 74 bytes captured (592 bits
    Section number: 1
  > Interface id: 0 (\Device\NPF_{1BF92210-918F-4051-B7F1-0BFE5C7FA703
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr  4, 2025 11:28:55.353121000 Hora de verano roman
    UTC Arrival Time: Apr  4, 2025 09:28:55.353121000 UTC
    Epoch Arrival Time: 1743758935.353121000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.039853000 seconds]
```

```
0000  90 09 df d1 1b e5 dc f8  b9 a1 b6 4b 08 00 45 00   ........ ...K..E.
0010  00 3c b1 d1 00 00 33 01  d6 d8 c1 92 7b 53 c0 a8   .<....3. ....{S..
0020  01 89 00 00 55 4e 00 01  00 0d 61 62 63 64 65 66   ....UN.. ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                    wabcdefg hi
```

```
Frame 90688: 74 bytes on wire (592 bits), 74 bytes captured (592 bits
    Section number: 1
  > Interface id: 0 (\Device\NPF_{1BF92210-918F-4051-B7F1-0BFE5C7FA703
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr  4, 2025 11:28:56.322727000 Hora de verano roman
    UTC Arrival Time: Apr  4, 2025 09:28:56.322727000 UTC
    Epoch Arrival Time: 1743758936.322727000
    [Time shift for this packet: 0.000000000 seconds]
```

```
0000  dc f8 b9 a1 b6 4b 90 09  df d1 1b e5 08 00 45 00   .....K.. ......E.
0010  00 3c 41 98 00 00 80 01  00 00 c0 a8 01 89 c1 92   .<A..... ........
0020  7b 53 08 00 4d 4d 00 01  00 0e 61 62 63 64 65 66   {S..MM.. ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                    wabcdefg hi
```

```
Frame 90689: 74 bytes on wire (592 bits), 74 bytes captured (592 bits
    Section number: 1
  > Interface id: 0 (\Device\NPF_{1BF92210-918F-4051-B7F1-0BFE5C7FA703
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr  4, 2025 11:28:56.362327000 Hora de verano roman
    UTC Arrival Time: Apr  4, 2025 09:28:56.362327000 UTC
```

```
0000  90 09 df d1 1b e5 dc f8  b9 a1 b6 4b 08 00 45 00   ........ ...K..E.
0010  00 3c bd d1 00 00 33 01  ca d8 c1 92 7b 53 c0 a8   .<....3. ....{S..
0020  01 89 00 00 55 4d 00 01  00 0e 61 62 63 64 65 66   ....UM.. ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                    wabcdefg hi
```

```
193.146.123.83    ICMP
192.168.1.137     ICMP
193.146.123.83    ICMP
192.168.1.137     ICMP
```

Conclusión:

Al capturar los pings a hosts remotos, observé que las direcciones MAC mostradas corresponden a los routers intermedios y no a los servidores remotos. Esto ocurre porque las direcciones MAC solo se usan dentro de la red local, mientras que los pings a hosts remotos pasan por varios dispositivos de red.

3: Lo curioso en estos datos:

La **dirección MAC** mostrada para estos hosts remotos generalmente no es la dirección MAC del servidor real (por ejemplo, el de www.cisco.com o www.google.com), sino de un **dispositivo de red intermedio** como un **router** o un **gateway**. Esto se debe a que las direcciones MAC solo se utilizan dentro de la red local, y los paquetes ICMP viajan a través de múltiples dispositivos de red en el camino hacia el host remoto.