**Práctica 1: Análisis de Protocolos con Wireshark**

***Autor****: Juan Rodríguez Suárez,* [*alu0101477596@ull.edu.es*](mailto:alu0101477596@ull.edu.es)*.*

***Curso:*** *Redes y Sistemas Distribuidos, 2º Grado en Ingeniería Informática.*

***Fecha:*** *3 de marzo de 2023.*

**Índice**

[1. Introducción 3](#_Toc129402697)

[2. Protocolos de la capa de enlace 4](#_Toc129402698)

[2.1. Protocolo Ethernet 4](#_Toc129402699)

[2.2. Protocolo ARP 5](#_Toc129402700)

[3. Protocolos de la capa de red 6](#_Toc129402701)

[3.1. Protocolo IP 6](#_Toc129402702)

[3.2. Protocolo ICMP 7](#_Toc129402703)

[4. Protocolos de la capa de transporte 8](#_Toc129402704)

[4.1. Protocolo UDP 8](#_Toc129402705)

[4.2. Protocolo TCP 9](#_Toc129402706)

[5. Protocolos de la capa de aplicación 10](#_Toc129402707)

[5.1. Protocolo DNS 10](#_Toc129402708)

[5.2. Protocolo HTTP 11](#_Toc129402709)

[6. Reflexión sobre los aspectos éticos de la utilización de Wireshark 12](#_Toc129402710)

[7. Referencias 13](#_Toc129402711)

# Introducción

A continuación, se redactan diferentes protocolos empleados en las comunicaciones entre sistemas distribuidos y en red. Todos ellos contribuyen a una comunicación entre dos o más dispositivos de una manera eficiente, segura y optimizada. Se distinguen dentro del modelo OSI cuatro capas muy importantes en el análisis de protocolos: capa de enlace (Ethernet, ARP); capa de red (IP, ICMP); capa de transporte (UDP, TCP); capa de aplicación (DNS, HTTP).

Para la realización del análisis de los protocolos mencionados anteriormente se usará la herramienta Wireshark, que es una herramienta de análisis de protocolos de red gratuita y de código abierto que se utiliza para capturar y analizar el tráfico de red en tiempo real. Es una herramienta muy útil para la solución de problemas de red, el monitoreo de redes y la seguridad de red. Wireshark permite a los usuarios capturar y examinar el tráfico de red en detalle, lo que significa que puede ver el contenido de los paquetes individuales que se envían y reciben en la red. La herramienta proporciona una interfaz gráfica de usuario (GUI) que muestra información detallada sobre el tráfico de red, como los protocolos utilizados, las direcciones IP de origen y destino, los puertos utilizados y el contenido de los datos.

Además de Wireshark, se hará uso de diferentes comandos de Linux muy útiles en el análisis del tráfico de una red. Algunos de estos comandos son *ping*, *traceroute*, *ifconfig*, etc

El hecho de entender cómo funcionan y de qué se encargan cada uno de estos protocolos de red es de una importancia mayor en el trabajo de un ingeniero de sistemas puesto que hoy en día la grandísima mayoría de dispositivos están conectados a una red con muchos otros dispositivos y es de vital importancia saber cómo es posible la interacción entre los diversos dispositivos además de otros aspectos claves como la seguridad e integridad del dispositivo en la red.

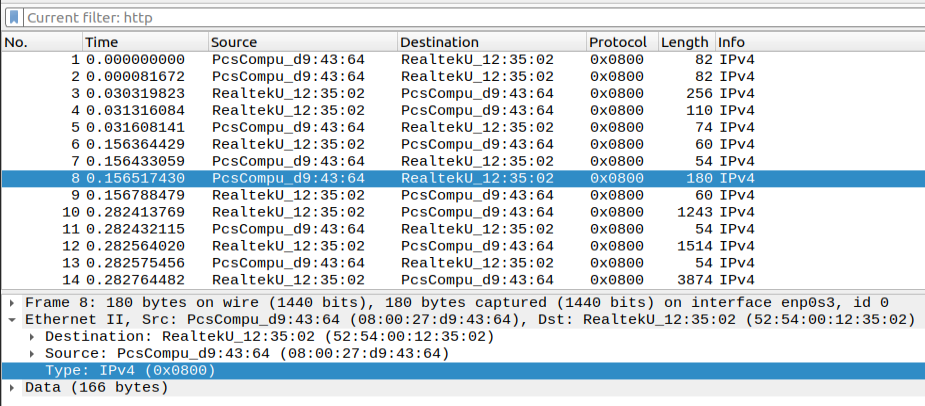
# Protocolos de la capa de enlace

## Protocolo Ethernet

Ethernet es un protocolo de red de nivel de enlace de datos que se utiliza comúnmente en las redes de área local (LAN). El protocolo Ethernet se utiliza para enviar paquetes de datos entre dispositivos en una red local. Los paquetes de datos se transmiten como tramas, que contienen distintos campos como la dirección de origen y la dirección de destino de los dispositivos, así como los datos que se están transmitiendo.

Esto último se puede observar en Wireshark, en la petición HTTP, donde en mi caso la dirección de destino MAC es 52:54:00:12:35:02 (con dispositivo RealtekU\_12) y la de origen es 08:00:27:d9:43:64 (con dispositivo PcsCompu\_d9). En cuanto al paquete de respuesta, se observa que la principal relación entre éste y el de petición es que su dirección de destino es la dirección local (la de PcsCompu) y la dirección de origen es la del servidor (RealtekU), lo que da pistas de que la comunicación ha funcionado correctamente.

Por otra parte, es interesante saber cuál es el protocolo de los datos que está transmitiendo Ethernet. Esto lo podemos saber gracias al campo Frame Type o Protocol, que indica el campo en la trama Ethernet que identifica el protocolo de capa superior que se está utilizando. En este caso, este campo es 0x0800 que se corresponde con el protocolo IPv4.

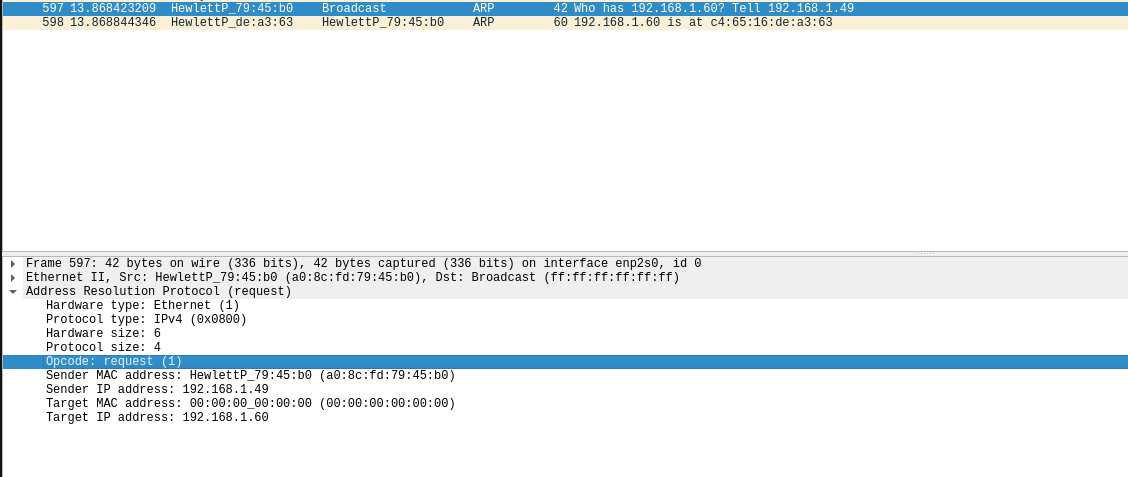


En conclusión, Ethernet es uno de los protocolos de red más utilizados y ampliamente aceptados en todo el mundo. Es una tecnología escalable, fácil de implementar y muy confiable, lo que lo hace ideal para su uso en redes locales y de área amplia.

## Protocolo ARP

El protocolo ARP (Address Resolution Protocol) es un protocolo a nivel de enlace cuya principal funcionalidad es resolver las direcciones IP a direcciones del nivel de enlace (MAC) de las máquinas de destino en redes locales. Funciona de la siguiente manera: Cuando un dispositivo desea comunicarse con otro dispositivo emitiendo una solicitud de ARP (ARP request), que contiene la dirección IP del dispositivo de destino. Esta solicitud se difunde en toda la red local y es recibida por todos los dispositivos en la red. El dispositivo de destino recibe la solicitud de ARP y responde con una respuesta de ARP (ARP reply) que contiene su dirección MAC correspondiente a la dirección IP que se proporcionó en la solicitud. Esta respuesta se envía directamente al dispositivo que solicitó la información. La estructura de ARP consta de los siguientes campos: Tipo de hardware, Tipo de protocolo, Longitud de la dirección de hardware, Longitud de la dirección de protocolo, Código de operación, Dirección MAC del remitente, Dirección IP del remitente, Dirección MAC del destinatario, Dirección IP del destinatario.

Como se puede observar, para la práctica, empleé mi dispositivo local con MAC A0:8C:FD:79:45:B0 e IP 192.168.1.49 y una impresora con MAC C4:65:16:DE:A3:63 e IP 192.168.1.60. Ambos dispositivos están conectados en una red local (la de mi casa). Cabe decir que el dispositivo que emite la solicitud está conectado a un switch en donde también está la impresora conectada.

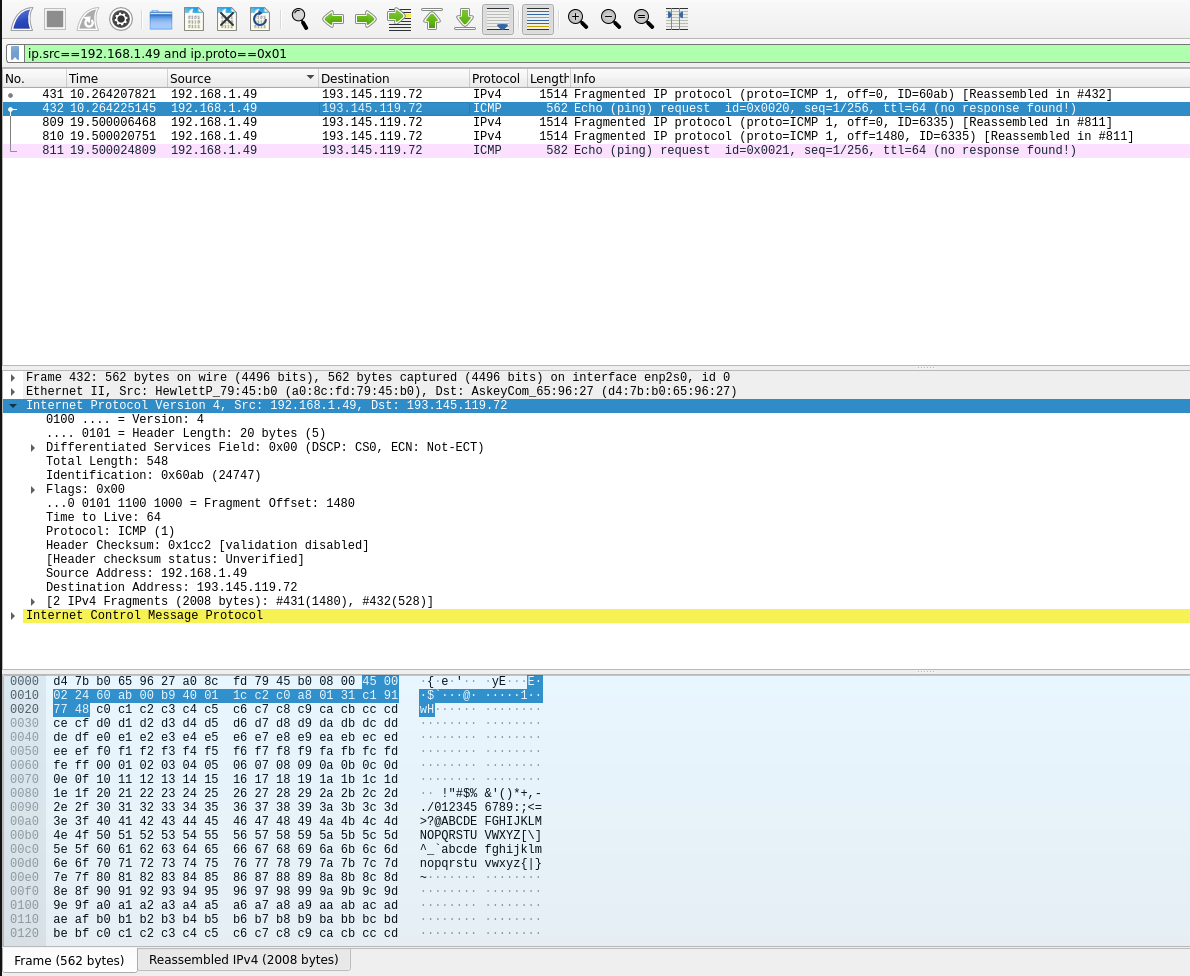


En este caso, el campo Frame Type de la trama Ethernet tiene valor 0x0806 (ARP) en contraposición al 0x0800 (IP) del apartado anterior. Esto se debe a las diferencias en el tipo de datos que lleva cada trama. El valor del Opcode es de 1, que se corresponde a la trama de petición. Por otra parte, en la respuesta la dirección de destino es la del equipo local y la de origen es la de la impresora. Si se envía un *ping* a un host de Internet, al ser ARP un protocolo para redes locales, solo se va a resolver una dirección; la de la puerta de enlace o rúter. La dirección IP de origen y destino seguirá siendo la misma y lo que irá cambiando entre enlaces y enlaces es la dirección MAC de origen y de destino. Por esta razón solo aparece en Wireshark un solo paquete ARP.

# Protocolos de la capa de red

## Protocolo IP

El protocolo IP (Internet Protocol) es un protocolo de red que se utiliza para enrutar paquetes de datos a través de redes de computadoras interconectadas. El protocolo IP es un protocolo sin conexión, lo que significa que no se establece una conexión antes de enviar datos. En su lugar, los datos se dividen en paquetes y se envían a través de la red individualmente. Cada paquete contiene información de encabezado que indica su origen, destino, longitud y otros datos necesarios para enrutar el paquete a través de la red.

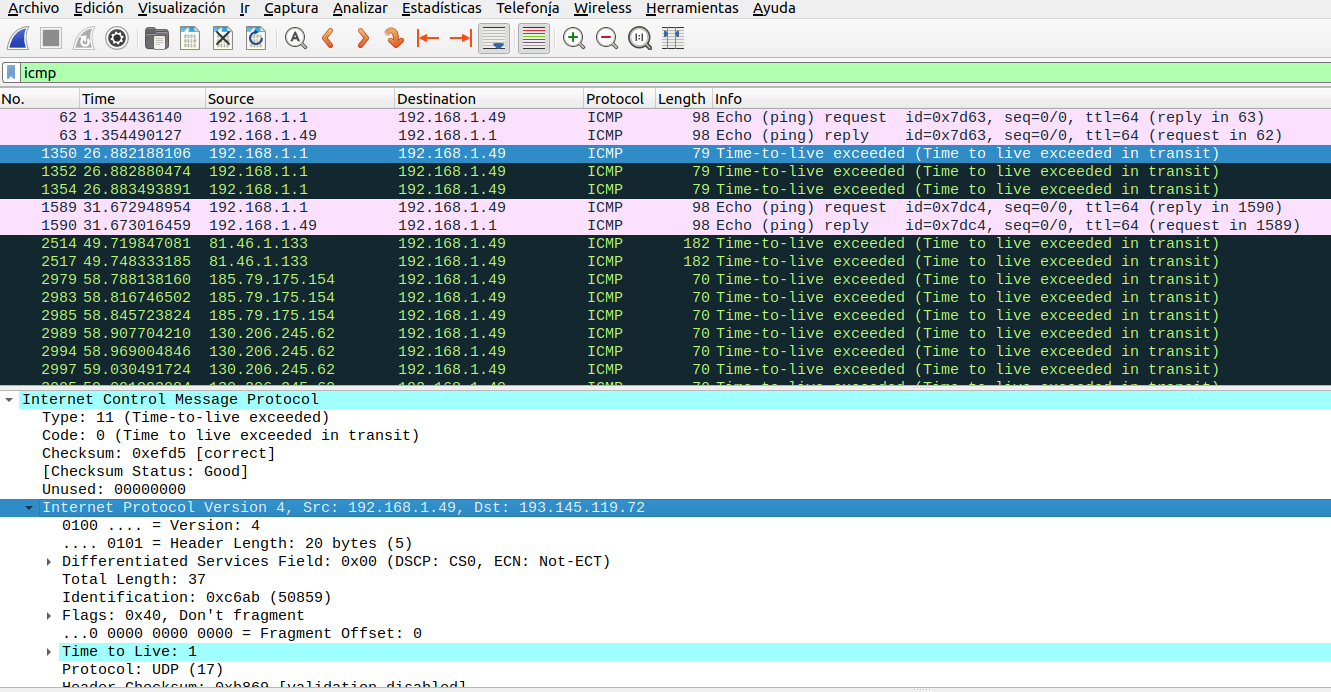


En la práctica, al enviar el paquete de 50 bytes se observa que mi dirección IP es 192.168.1.48 y que el campo de protocolo es 0x01 (ICMP). Hay 20 bytes de cabecera y una longitud total de 78 bytes, por tanto, hay 58 bytes de datos (50 si descontamos la cabecera de ICMP). En cuanto a los otros dos paquetes de 2000 y 3500 bytes, para saber si han sido fragmentados se utiliza el campo flags, que indica si el paquete ha sido fragmentado (0x20). Para el primero se mandan dos fragmentos de 1480 y 548 bytes cuyo identificador es el del paquete (0x60ab) y cada fragmento tiene un offset distinto. Para el segundo se mandan tres fragmentos de 1480, 1480 y 548 bytes cuyo id. es 0x6335 y offset distintos que indican el orden dentro del paquete. Se sabe que para el primer paquete son dos fragmentos porque el segundo fragmento no tiene más flags y se puede saber qué fragmento es dentro del paquete gracias al offset. Por tanto, los campos que cambian en cada fragmento son el offset y los flags dependiendo de si es el último o no (el de identificador es el mismo porque pertenecen al mismo paquete).

## Protocolo ICMP

El protocolo ICMP es un protocolo de nivel de red que se utiliza para enviar mensajes de control y error en la red. ICMP se utiliza para proporcionar información de diagnóstico y control de tráfico en la red, como informar de errores en la entrega de paquetes, medir la latencia de la red y verificar la conectividad de los dispositivos de red. Los mensajes ICMP son enviados en paquetes IP y su función principal es proporcionar información sobre el estado y la salud de la red. Estos mensajes son generados por dispositivos de red como routers, firewalls, switches, etc. Algunos de los mensajes ICMP más comunes son: Echo Request y Echo Reply.

En la práctica, con referencia al comando ping, envié un mensaje desde el equipo local (192.168.1.49) a otra máquina 192.168.1.60. Al analizar uno de los paquetes request mandados por el comando ping, se observa que el campo tipo es 8 y el código es 0 (echo (ping) request). También hay otros campos como la suma de comprobación, el identificador y la secuencia que ocupan 2 bytes cada uno. En el paquete reply es prácticamente idéntico excepto por tipo que es 0 y el código es 0 (echo (ping) reply). También ha cambiado el valor de la suma de comprobación.

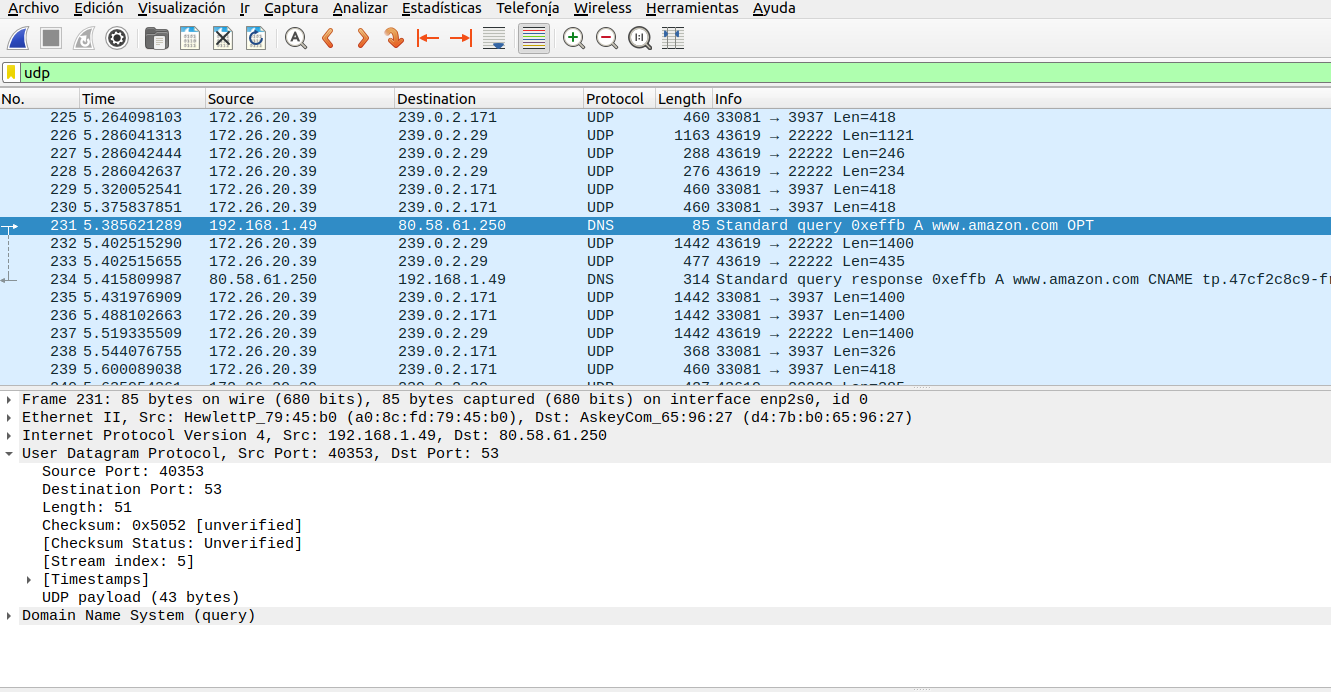
 Para comprobar los paquetes de traceroute, usé la misma máquina y usé traceroute con www.etsii.ull.es (193.145.119.72) y se observa que en paquetes enviados sucesivos el TTL se incrementa en 1. Esto es debido a que traceroute envía un mensaje con TTL 1, lo recibe por el primer rúter que lo ha devuelto por haber agotado su TTL y lo vuelve a mandar, pero con el TTL ahora en 2 y así hasta que llegue. En los paquetes enviados por traceroute, en el campo de protocolo se observa que es UDP. La razón principal se debe a una limitación de ICMP. Los paquetes ICMP están sujetos a ciertas restricciones en muchos sistemas operativos y dispositivos de red, que pueden bloquear o limitar la velocidad de los paquetes ICMP entrantes o salientes. Sin embargo, los paquetes devueltos por los routers son de tipo 11 (TTL exceeded) excepto el devuelto por el nodo final que tiene tipo y código 3: Destination unreachable (Port unreachable). Son diferentes porque una vez ha llegado al destino, la carga útil es un datagrama UDP dirigido a un puerto aleatorio y como lo más probable es que el puerto esté cerrado, reenvía un mensaje ICMP con el anterior código de error.

# Protocolos de la capa de transporte

## Protocolo UDP

El protocolo UDP (User Datagram Protocol) es un protocolo de comunicaciones en redes de computadoras que se utiliza para enviar y recibir datos sin establecer una conexión previa. A diferencia del protocolo TCP (Transmission Control Protocol), que proporciona una comunicación orientada a la conexión y garantiza la entrega de los datos en el orden correcto, UDP no proporciona garantías de entrega, orden o control de flujo. En su lugar, UDP es un protocolo más ligero y eficiente que se utiliza principalmente para aplicaciones que requieren una transmisión de datos rápida y en tiempo real, como los juegos en línea o las transmisiones de video y audio en directo.

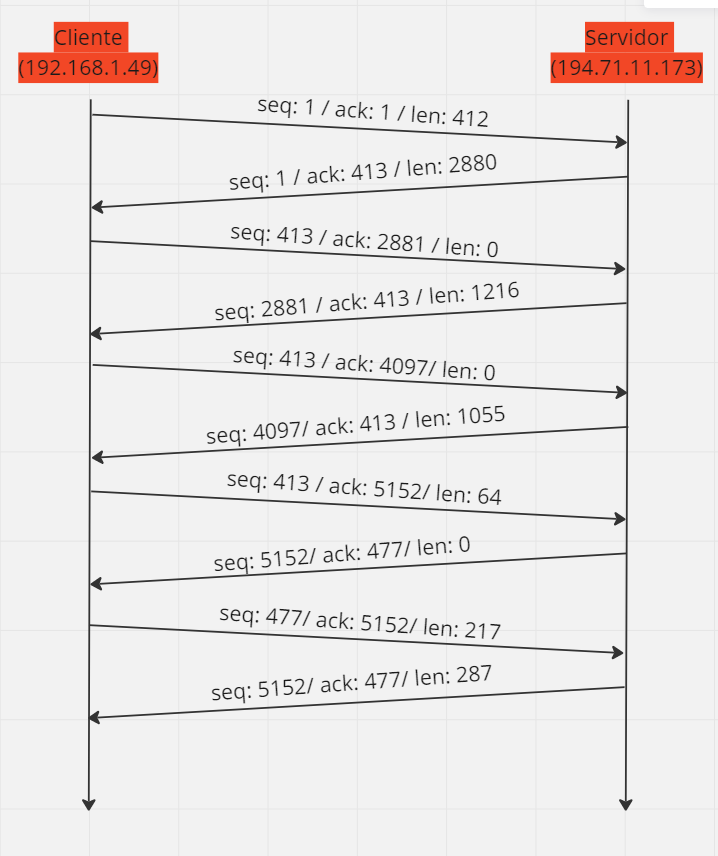
En relación a la práctica, ejecuté el comando “dig www.amazon.com” y capturé los paquetes UDP enviados y recibidos. Se observa que el campo longitud en el enviado es de 51 bytes y su carga es de 43 bytes, por tanto, el campo longitud se refiere al tamaño del segmento completo incluyendo la cabecera de 8 bytes. El número máximo de bytes de la carga útil sería el número que se puede representar con los 2 bytes del campo longitud, que es 216 = 65536 bytes – 8 bytes de la cabecera = 65528 bytes de carga útil máxima. Por tanto, siguiendo la misma lógica, el mayor puerto de origen que puede contener es 216 = 65536 y su número de protocolo es 17 (0x11). Por otra parte, la suma de comprobación se realiza sobre los siguientes campos del encabezado UDP: Puerto de origen, Puerto de destino, Longitud, Suma de comprobación (a 0) y los Datos de la carga útil.

 En cuanto a la relación de los puertos de origen y destino de los paquetes enviados y recibidos, se observa que los puertos están cambiados de orden entre el paquete enviado y recibido (40353 es el puerto local y 53 el de envío):

## Protocolo TCP

El protocolo TCP es similar al UDP pero este se usa para conexiones seguras en donde se necesitan todos los datos de manera fiable y completa. Además, requiere establecer una conexión previamente con el destino antes de enviar los datos. En la relación a la práctica, se observa que el número de secuencia del segmento TCP SYN es 0 (se sabe porque en la parte de la cabecera Flags está marcado el penúltimo bit que se refiere a SYN). Estos números de secuencia se generan con la expresión: ISN = M + F(localip, localport, remoteip, remoteport, secretkey) donde M es el temporizador de 4 microsegundos, y F() es una función pseudoaleatoria. Esto es así por motivos de seguridad principalmente y evitar que un atacante pueda interceptar los segmentos fácilmente. De forma análoga, tenemos el segmento de respuesta SYN-ACK con número de secuencia 0 y se sabe que es un SYN-ACK porque en el campo Flags está marcado el segundo y quinto bit que se corresponden con Acknowledgment y Syn respectivamente. El tercer segmento se corresponde con el ACK de respuesta del cliente que completa la secuencia de inicio, que no contiene carga útil. Tanto el segmento inicial SYN como el SYN-ACK contienen opciones como Maximum segment size, SACK permitted, Timestamps o No-Operation que se utilizan para proporcionar información adicional sobre la conexión o para ajustar ciertos parámetros de la comunicación TCP.

A continuación, se presenta una traza de los 10 primeros paquetes después del a fase de conexión entre el cliente y el servidor donde se destaca el número de secuencia, el ACK y la longitud del segmento. Parecer ser que en el último paquete no se recibieron los datos correctamente porque el ACK es 477 cuando debería ser 477 + 217



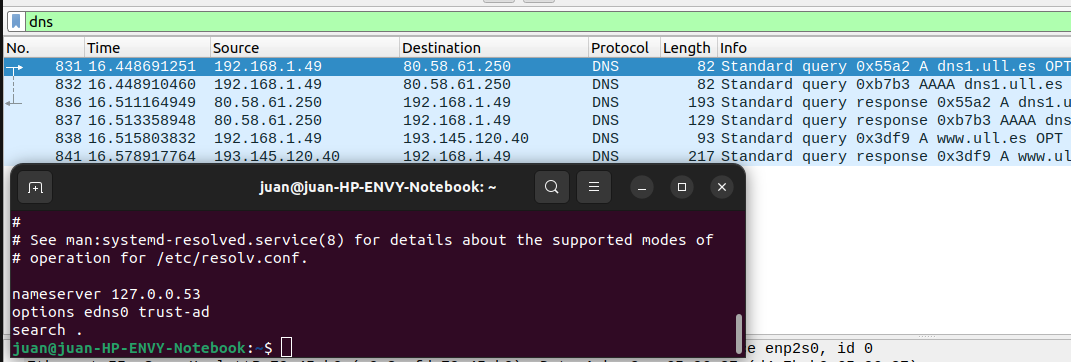
# Protocolos de la capa de aplicación

## Protocolo DNS

El protocolo DNS se utiliza para resolver los nombres de dominio en direcciones IP. Así, con este protocolo se traducen los nombres de hosts como www.etsii.ull.es en la dirección IP 193.145.119.72. En la práctica, si hacemos “dig www.ulpgc.es.”, se observa que en el primer paquete DNS el puerto de destino es el 53, que es el bien conocido por el protocolo DNS. También se ve que su dirección de destino es 80.58.61.250, que usando el comando “nslookup 80.58.61.250” nos dice que su nombre de host es “250.red-80-58-61.staticip.rima-tde.net.”. Usando el comando “resolvectl status” sabemos que el servidor DNS actual primario es esa dirección IP mencionada anteriormente, por lo que parece que está preguntando directamente al servidor DNS principal. Cabe destacar que esta dirección IP no aparece en “/etc/resolv.conf”. El mensaje de petición es de Tipo A (Host Address) que se refiere a que se busca resolver el nombre de www.ulpgc.es a una dirección IPv4. Existen otros tipos como AAAA para IPv6, MX para correo electrónico o NS para los servidores de nombres. Por otro lado, si nos fijamos en el paquete de respuesta, contiene dos respuestas: una con “www.ulpgc.es” que es un alias de la segunda respuesta que es “cluster-web.ulpgc.es” que, entre otras muchas cosas, contiene su dirección IP 193.145.138.32. Estas respuestas contienen también el tipo (que en este caso es A, que se refiere a una dirección IPv4), el TTL y la clase que es IN (consulta a Internet).

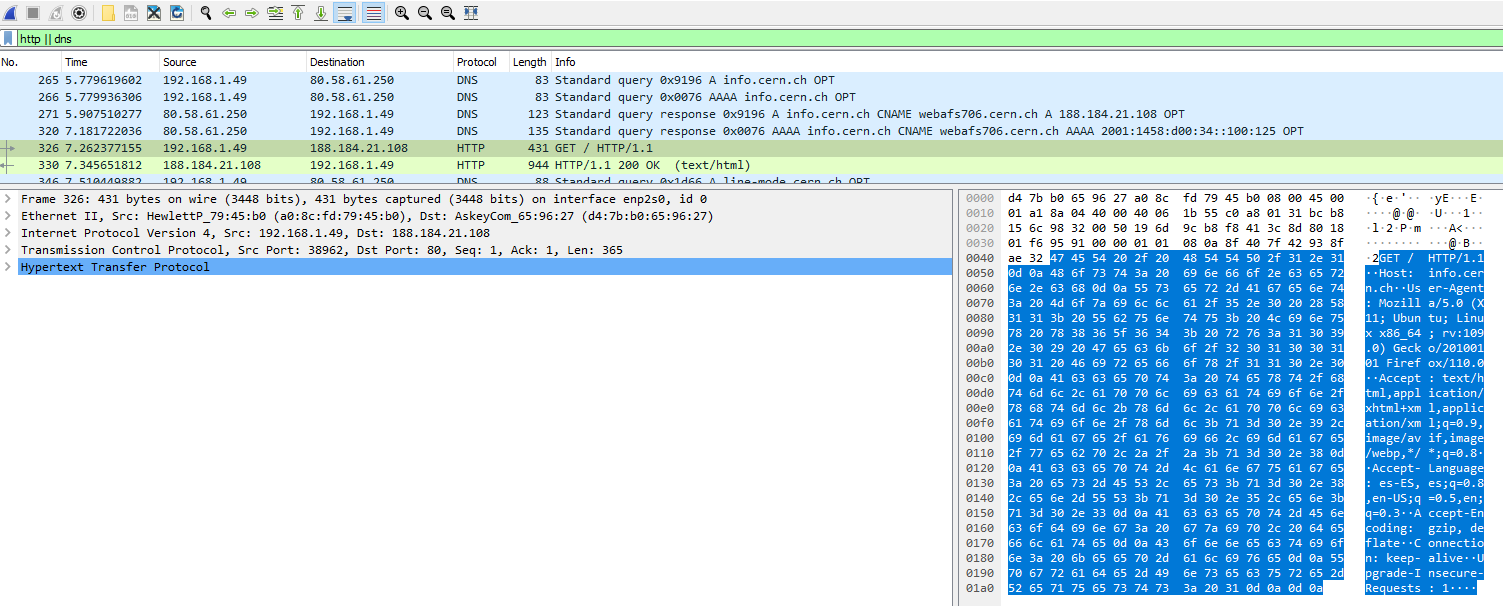
Si preguntamos por un servidor de nombres, el puerto de destino sigue siendo 53, que es el bien conocido por DNS, la dirección IP de destino y host son los mismos que antes y el mensaje de petición es de tipo NS (Name Server) que se refiere a un servidor de nombres. Ahora, el mensaje de respuesta contiene dos respuestas que se corresponden a dos NS: chico.rediris.es y sun.rediris.es que contienen el tipo (NS), clase (IN), TTL y la longitud de los datos.

Si preguntamos a un servidor de nombres alternativo, se hacen tres peticiones: dos para resolver la IPv4 e IPv6 de dns1.ull.es y otra para resolver www.ull.es. Esto es así porque primero necesita resolver a qué dirección se refiere dns1.ull.es y luego resolver www.ull.es. De nuevo se observa que el archivo /etc/resolv.conf no contiene ninguna de las direcciones de destino indicadas y sigue utilizando el DNS predeterminado mencionado anteriormente.



## Protocolo HTTP

El protocolo HTTP (Hypertext Transfer Protocol) se utiliza para la comunicación entre clientes (como navegadores web) y servidores web. Se utiliza para solicitar y enviar información, como páginas web, imágenes, videos y otros recursos de la web. En la práctica, al abrir la página web “http://info.cern.ch” primero se necesita convertir ese nombre a una dirección IP válida, por eso lo primero que se ve observa es una petición al servidor DNS para resolver ese nombre. Posteriormente envía una petición HTTP GET en la que se ve que el navegador (Firefox) está usando la versión 1.1 del protocolo y que el servidor acepta los lenguajes español e inglés. Por otro lado, si miramos el mensaje de respuesta vemos que el servidor usa la versión 1.1 y devuelve un código de estado de 200 (OK). Otro código de estado muy conocido es el 404 o Not Found que se ha popularizado en campos más allá de la informática y las redes. Estos códigos indican el estado de la solicitud realizada por un cliente al servidor web. Los códigos generalizados son: 1xx – Información, 2xx – Éxito, 3xx – Redirección, 4xx – Error del cliente, 5xx – Error del servidor. Continuando con el mensaje de respuesta, también dice que la última vez que se modificó la página fue el 5 de febrero de 2014 a las 16:00 GMT. Otra información clave es que el servidor ha devuelvo 646 bytes de contenido.

 Una vez se ha recargado la página, si observamos la primera petición (la de antes) no se ve ninguna línea If-Modified-Since pero en este último GET sí que hay una que indica la fecha de la última modificación (la fecha de antes). A esta línea le sigue otra If-None-Match que cuando la revise el servidor, verificará si el valor de la cabecera coincide con el valor de la etiqueta de entidad (etag) del recurso solicitado. Si coinciden, significa que la versión del recurso no ha cambiado desde la última vez que se solicitó, por lo que el servidor responderá con un código de estado "304 Not Modified" (código devuelto en la práctica), indicando que el recurso no ha sido modificado y que el cliente puede utilizar su copia en caché. En caso contrario, el servidor enviará una respuesta normal con un código de estado "200 OK" y los datos del recurso solicitado. Por tanto, el servidor no devuelve ningún fichero HTML porque la página no ha sido modificada. Cambiando de tema, en los paquetes HTTP, al ser un protocolo textual, son legibles por el ser humano porque están codificados en ASCII, sin embargo, en Wireshark parece que solo es legible a partir de cierto punto. Esto se debe a que el paquete entero no solo es HTTP, sino que está formado por muchos otros protocolos binarios (abstracción). En la captura se observa que estos protocolos son: Ethernet, IP, TCP y finalmente HTTP (ver imagen y la ‘frontera’ entre el protocolo HTTP y TCP):

# Reflexión sobre los aspectos éticos de la utilización de Wireshark

El uso de Wireshark también puede plantear preocupaciones éticas que deben ser consideradas y abordadas. En particular, se deben respetar la privacidad y los derechos de los usuarios de la red durante el uso de la herramienta.

Primero, es importante obtener el consentimiento de los usuarios antes de analizar su tráfico de red con Wireshark. Si se utiliza la herramienta para interceptar y examinar el tráfico de red sin el consentimiento de los usuarios, se estaría violando su privacidad y sus derechos. Por lo tanto, es importante obtener el consentimiento de los usuarios antes de analizar su tráfico de red.

Además, durante el análisis del tráfico de red, se pueden capturar información confidencial, como contraseñas o información personal. Es importante tratar esta información con cuidado y confidencialidad para evitar su divulgación no autorizada. Se deben tomar medidas para garantizar que cualquier información confidencial capturada durante el análisis del tráfico de red esté debidamente protegida y que solo se use con el propósito previsto.

Otro aspecto ético importante es el uso malintencionado de Wireshark. Si se utiliza la herramienta para espiar a otros usuarios de la red, como para obtener información confidencial o para espiar su actividad en línea, se estaría cometiendo una violación clara de la privacidad y los derechos de las personas. Cualquier uso de Wireshark con fines malintencionados debe ser evitado a toda costa.

En general, es importante utilizar la herramienta de manera ética y responsable. Esto implica obtener el consentimiento de los usuarios antes de analizar su tráfico de red, tomar medidas para garantizar la privacidad y confidencialidad de cualquier información capturada durante el análisis, y evitar el uso malintencionado de la herramienta.

En conclusión, el uso ético de Wireshark es fundamental para respetar la privacidad y los derechos de los usuarios de la red. Los profesionales de las redes y la informática deben utilizar la herramienta con precaución y tratar cualquier información confidencial con cuidado y confidencialidad. Wireshark no debe ser utilizado para actividades malintencionadas que puedan violar la privacidad de otras personas.

# Referencias

-Wireshark: <https://www.wireshark.org>

-Cisco: <https://www.cisco.com>

-IETF: <https://www.ietf.org>

-RFC: <https://www.rfc-editor.org>