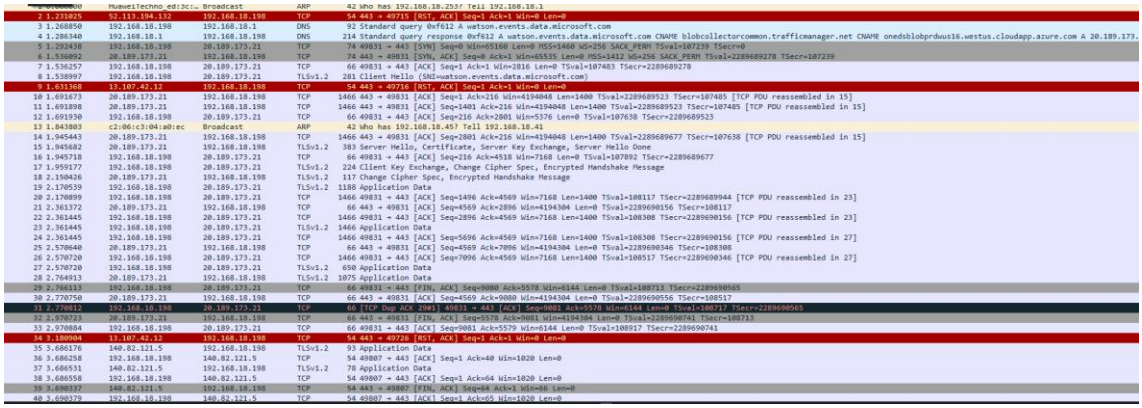
Fundamentos de Redes de Datos

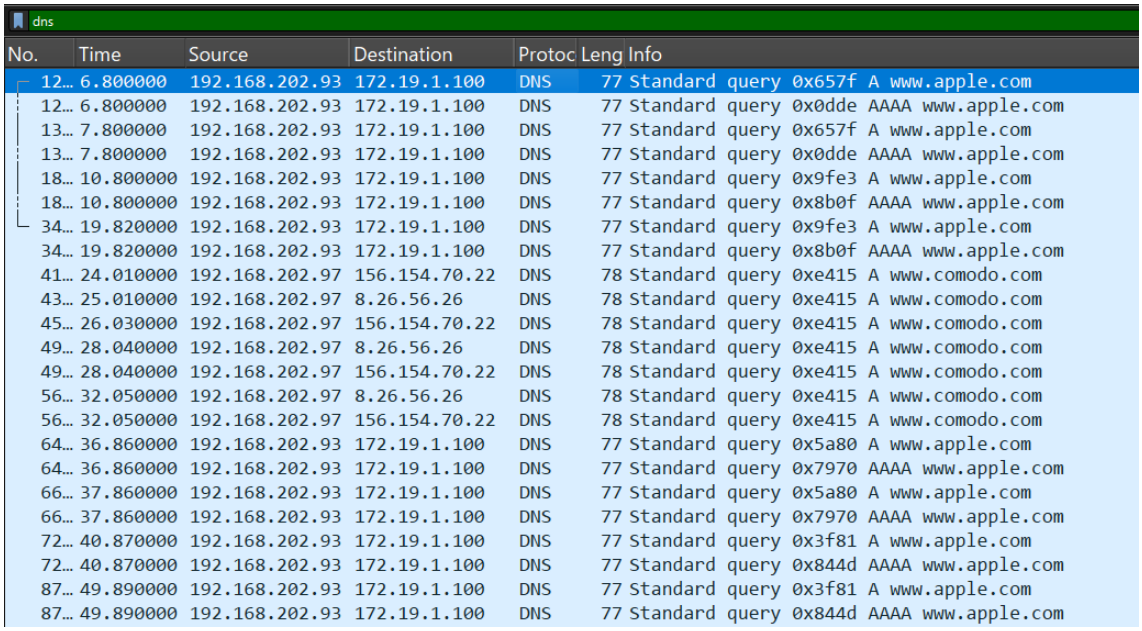
Tarea 2: Arquitectura de Red

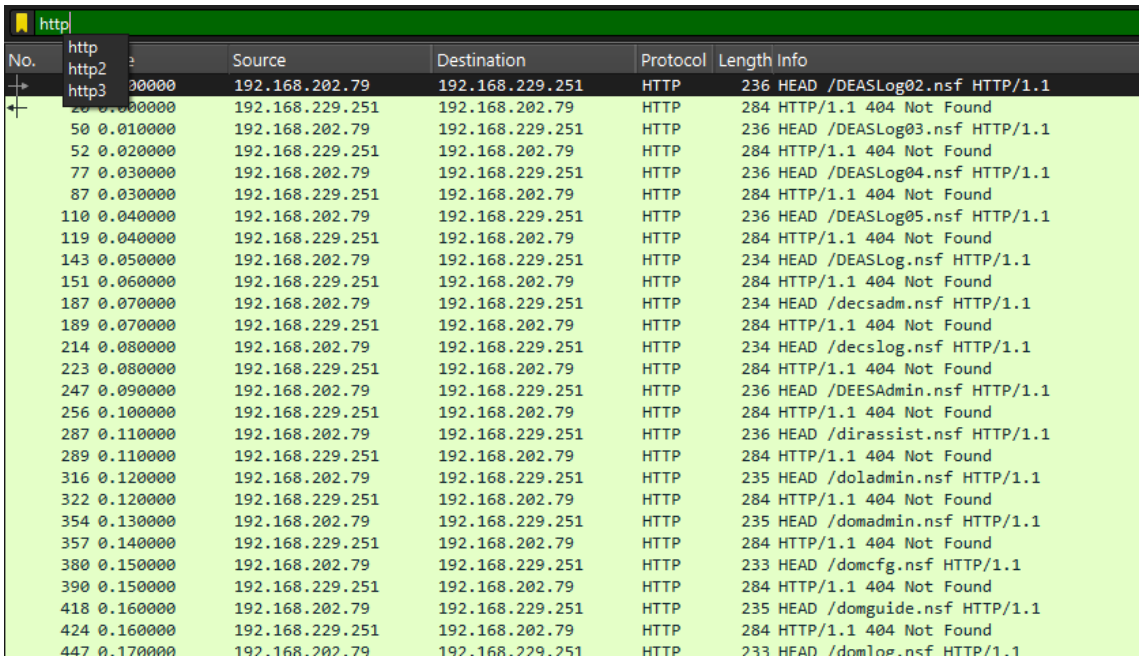
Utiliza el programa *Wireshark* para capturar el tráfico de red en el interfaz principal de un computador cualquiera (puedes hacerlo en un laboratorio, tu portátil, tu PC de casa, o cualquier otro en el que esté instalado dicho programa). Asegúrate de que, durante el periodo de captura, se captura al menos tráfico correspondiente a una consulta DNS, así como una petición de una página web a un servidor cualquiera.

1. Especifica los comandos / programas utilizados durante la captura para asegurarte de que se generaban tramas con los dos tipos de tráfico (DNS y HTTP) solicitados.



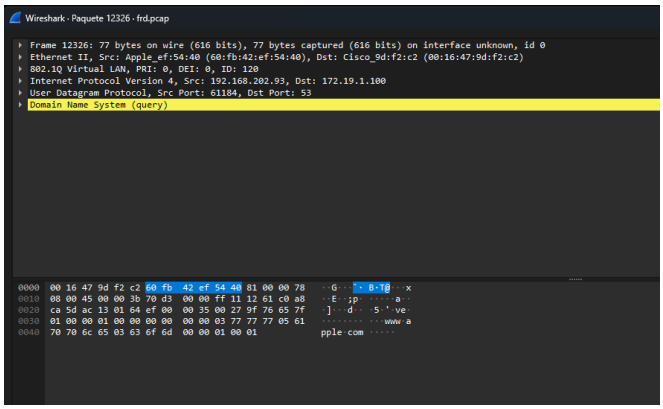
En la barra de búsqueda de Wireshark puedo determinar el tipo de protocolo que quiero a modo de filtrado.





No.	Time	Source	Destination	Protocol	Length	Info
	00000	192.168.202.79	192.168.229.251	HTTP	236	HEAD /DEASLog02.nsf HTTP/1.1
	000000	192.168.229.251	192.168.202.79	HTTP	284	HTTP/1.1 404 Not Found
50	0.010000	192.168.202.79	192.168.229.251	HTTP	236	HEAD /DEASLog03.nsf HTTP/1.1
52	0.020000	192.168.229.251	192.168.202.79	HTTP	284	HTTP/1.1 404 Not Found
77	0.030000	192.168.202.79	192.168.229.251	HTTP	236	HEAD /DEASLog04.nsf HTTP/1.1
87	0.030000	192.168.229.251	192.168.202.79	HTTP	284	HTTP/1.1 404 Not Found
110	0.040000	192.168.202.79	192.168.229.251	HTTP	236	HEAD /DEASLog05.nsf HTTP/1.1
119	0.040000	192.168.229.251	192.168.202.79	HTTP	284	HTTP/1.1 404 Not Found
143	0.050000	192.168.202.79	192.168.229.251	HTTP	234	HEAD /DEASLog.nsf HTTP/1.1
151	0.060000	192.168.229.251	192.168.202.79	HTTP	284	HTTP/1.1 404 Not Found
187	0.070000	192.168.202.79	192.168.229.251	HTTP	234	HEAD /decsadm.nsf HTTP/1.1
189	0.070000	192.168.229.251	192.168.202.79	HTTP	284	HTTP/1.1 404 Not Found
214	0.080000	192.168.202.79	192.168.229.251	HTTP	234	HEAD /decslog.nsf HTTP/1.1
223	0.080000	192.168.229.251	192.168.202.79	HTTP	284	HTTP/1.1 404 Not Found
247	0.090000	192.168.202.79	192.168.229.251	HTTP	236	HEAD /DEESAdmin.nsf HTTP/1.1
256	0.100000	192.168.229.251	192.168.202.79	HTTP	284	HTTP/1.1 404 Not Found
287	0.110000	192.168.202.79	192.168.229.251	HTTP	236	HEAD /dirassist.nsf HTTP/1.1
289	0.110000	192.168.229.251	192.168.202.79	HTTP	284	HTTP/1.1 404 Not Found
316	0.120000	192.168.202.79	192.168.229.251	HTTP	235	HEAD /doladmin.nsf HTTP/1.1
322	0.120000	192.168.229.251	192.168.202.79	HTTP	284	HTTP/1.1 404 Not Found
354	0.130000	192.168.202.79	192.168.229.251	HTTP	235	HEAD /domadmin.nsf HTTP/1.1
357	0.140000	192.168.229.251	192.168.202.79	HTTP	284	HTTP/1.1 404 Not Found
380	0.150000	192.168.202.79	192.168.229.251	HTTP	233	HEAD /domcfg.nsf HTTP/1.1
390	0.150000	192.168.229.251	192.168.202.79	HTTP	284	HTTP/1.1 404 Not Found
418	0.160000	192.168.202.79	192.168.229.251	HTTP	235	HEAD /domguide.nsf HTTP/1.1
424	0.160000	192.168.229.251	192.168.202.79	HTTP	284	HTTP/1.1 404 Not Found
447	0.170000	192.168.202.79	192.168.229.251	HTTP	233	HEAD /domlog.nsf HTTP/1.1

2. Escoge una cualquiera de las tramas DNS capturadas y, adjuntando un pantallazo de ésta en el que aparezca claramente toda la información necesaria, contesta a los siguientes apartados:



a) ¿Qué protocolo de transporte encapsula al mensaje original DNS de la capa de aplicación? Especifica cuál es el tamaño exacto (en bytes) del mensaje a nivel de aplicación, así como el de la cabecera añadida por dicho protocolo.

El protocolo de transporte que encapsula el mensaje DNS es normalmente **UDP**. El tamaño exacto del mensaje DNS es de **77 bytes** y la cabecera UDP tiene un tamaño fijo de **8 bytes**.

b) ¿Dentro de qué protocolo de red viaja el anterior segmento? ¿Cuál es el tamaño en bytes añadido por la cabecera de este otro protocolo?

El segmento viaja dentro de la **IPv4**, con una cabecera de **20 bytes**.

c) ¿Cuál es el tamaño total de la trama Ethernet que encapsula al segmento anterior?

El tamaño total de la trama Ethernet es de **77 bytes**, donde se incluye la cabecera Ethernet (14 bytes), la cabecera IP (20 bytes), la cabecera UDP (8 bytes) y los datos del mensaje DNS.

d) Calcula la eficiencia de uso en % (es decir, el porcentaje de datos del nivel de aplicación enviados respecto al tamaño total final de la trama, que incluye todas las cabeceras de protocolos encapsulados comentados).

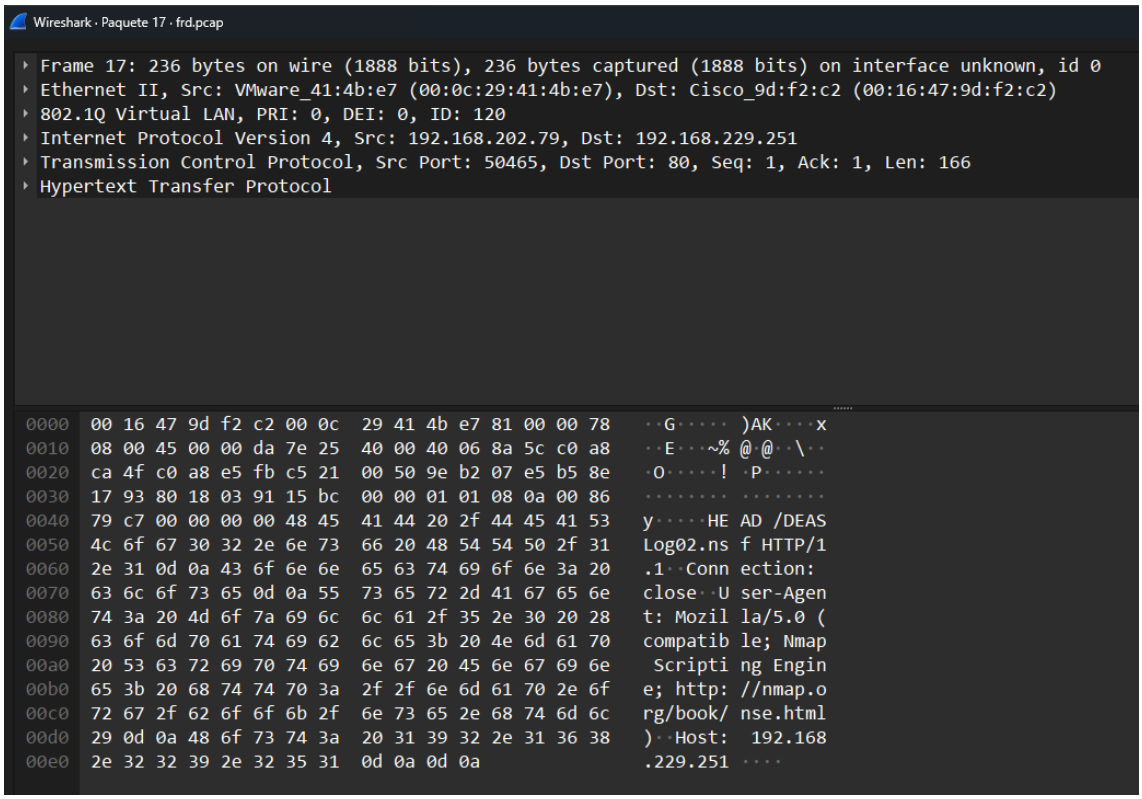
Para calcular la eficiencia de uso (%), utilizaremos la siguiente fórmula:

$$\text{Eficiencia} = \left(\frac{\text{Tamaño de los datos a nivel de aplicación}}{\text{Tamaño total de la trama}}\right) \times 100$$

- Datos a nivel de aplicación (DNS): 77 bytes
- Ethernet: 14 bytes
- IP: 20 bytes
- UDP: 8 bytes

$$\text{Eficiencia} = \left(\frac{77 - (14 + 20 + 8)}{77}\right) \times 100 = \left(\frac{35}{77}\right) \times 100 \approx 45.45\%$$

3. Repite el ejercicio anterior, con todos sus subapartados, pero en este caso para una trama cualquiera correspondiente al tráfico HTTP generado durante la captura.



a) El tráfico HTTP viaja encapsulado en **TCP**. En la captura se observa que el tamaño del mensaje es de **166 bytes,** pero la cabecera TCP tiene **20 bytes** (sin opciones adicionales).

b) El segmento viaja dentro de la **IP**, con una cabecera de **20 bytes**.
c) El tamaño total de la trama es de **236 bytes**. Este valor incluye la cabecera Ethernet (14 bytes), la cabecera IP, la cabecera TCP y los datos HTTP.
d) Para calcular la eficiencia de uso (%), utilizaremos la siguiente fórmula:

$$\text{Eficiencia} = \left(\frac{\text{Tamaño de los datos a nivel de aplicación}}{\text{Tamaño total de la trama}}\right) \times 100$$

- Datos a nivel de aplicación (DNS): 166 bytes
- Ethernet: 14 bytes
- IP: 20 bytes
- TCP: 20 bytes

$$\text{Eficiencia} = \left(\frac{166}{236}\right) \times 100 \approx 70.34\%$$

4. En función de los resultados que obtengas en las preguntas anteriores, ¿podrías afirmar cuál de los dos casos es más eficiente desde el punto de vista del porcentaje de datos útiles enviados?

   La trama HTTP es más eficiente desde el punto de vista del porcentaje de datos útiles enviados. Esto se debe a que, en el caso del tráfico HTTP, la proporción de datos de aplicación respecto al tamaño total de la trama es mayor, lo que implica un menor porcentaje de bytes destinados a las cabeceras.

5. Dada una arquitectura de red, reflexiona brevemente sobre las implicaciones derivadas de tener un número determinado de capas. Por ejemplo, ¿crees que a mayor número de capas siempre habrá un mayor número de bytes de cabecera?

   Un mayor número de capas tiende a aumentar la cantidad de bytes de cabecera, pero este es un **compromiso necesario** para garantizar la **funcionalidad**, **modularidad**, y **escalabilidad** de la red. Aunque este incremento en la sobrecarga puede reducir la eficiencia de transmisión en términos de bytes útiles, aporta beneficios importantes en términos de flexibilidad, facilidad de mantenimiento, y capacidad para adaptarse a diferentes entornos de red.

6. Finalmente, ¿crees que un router necesita acceder a los datos a nivel aplicación (por ejemplo a los mensajes HTTP) para hacer su trabajo? ¿Podrías citar algún tipo de analogía similar relacionada con el transporte de información (no necesariamente en Internet) para reafirmar tu respuesta?

   No, un router no necesita acceder a los datos del nivel de aplicación, como los mensajes HTTP, para hacer su trabajo. El router solo necesita información contenida en la cabecera de la capa de red (principalmente la dirección IP de destino), junto con algunas otras informaciones de control, para determinar el mejor camino para el paquete a través de la red.

   Una analogía adecuada para entender esto sería:

- Imagina que tienes un **servicio postal**. Supongamos que deseas enviar una carta desde una ciudad a otra.
- El **cartero** o **sistema postal** (el equivalente al router) solo necesita ver la **dirección** escrita en el sobre (similar a la dirección IP en la cabecera de la capa de red). El cartero no necesita abrir la carta para ver su contenido (equivalente a los datos de la aplicación como HTTP).
- El contenido de la carta es irrelevante para el cartero porque su trabajo es únicamente entregar el sobre al destinatario correcto basándose en la dirección.

De forma similar, el router se centra únicamente en la información de direccionamiento de los paquetes, sin preocuparse por el contenido de los mismos, lo cual está en capas más altas (como la capa de aplicación).