

**UNIVERSIDAD DE LOS ANDES  
DEPARTAMENTO DE INGENIERIA DE  
SISTEMAS Y COMPUTACIÓN**



**LABORATORIO 2: ANÁLISIS DE PROTOCOLOS DE  
LA CAPA DE APLICACIÓN**

**ISIS 3204 – INFRESTRUCTURA DE  
COMUNICACIONES**

**Profesor 1: Yuri Pinto  
Profesor 2: Nathalia Quiroga**

**Grupo 8  
Sofia Vasquez - 202123910  
Isabella Caputi - 202122075  
Giuliana Volpi - 202123986**

**2025-10**

## **Tabla de Contenido**

1.	Pruebas de Conectividad .....	3-7
2.	Análisis de tráfico del Servicio DNS .....	7-13
3.	Análisis de tráfico del Servicio FTP.....	13-16
4.	Análisis de tráfico del Servicio Web.....	16-18
5.	Análisis del protocolo HTTPS realizando navegación en el sitio de YouTube.....	19-20
6.	Análisis del protocolo VoIP.....	20-23
7.	Análisis del protocolo RTMP.....	23-25
8.	Bibliografía .....	25

## 1. Pruebas de Conectividad

**1.1 Prueba ping DNS:** Realice pruebas de conectividad desde el Cliente al Servidor seleccionado para prestar el servicio de DNS en la red, para esta prueba utilice la dirección IP del servidor. Guarde en un archivo la captura del tráfico con el nombre Ping\_DNS\_IP.pcap

Para comprobar la conectividad entre el cliente y el servidor encargado de prestar el servicio DNS en la red, se ejecutó el comando ping desde el equipo cliente hacia la dirección IP del servidor (**172.20.10.2**).

```
C:\Users\Sofia Toro>ping 172.20.10.2

Pinging 172.20.10.2 with 32 bytes of data:
Reply from 172.20.10.2: bytes=32 time=15ms TTL=64
Reply from 172.20.10.2: bytes=32 time=62ms TTL=64
Reply from 172.20.10.2: bytes=32 time=76ms TTL=64
Reply from 172.20.10.2: bytes=32 time=83ms TTL=64

Ping statistics for 172.20.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 15ms, Maximum = 83ms, Average = 59ms
```

Ilustración 1: Ping DNS con IP

En la Ilustración anterior se evidencia que, durante esta prueba, se enviaron 4 paquetes ICMP Echo Request y se recibieron 4 respuestas ICMP Echo Reply, lo que indica que existe comunicación bidireccional entre ambos equipos y no se presenta pérdida de paquetes. Los tiempos de respuesta oscilaron entre 15 ms y 83 ms, con un promedio de 59 ms, lo cual evidencia un retardo bajo y una conectividad estable.

Continuamos con la prueba de Wireshark:

No.	Time	Source	Destination	Protocol	Length	Info
62	6.948921	172.20.10.7	172.20.10.2	ICMP	74	Echo (ping) request id=0x0001, seq=17/4352, ttl=128 (no response found!)
63	6.948929	172.20.10.7	172.20.10.2	ICMP	74	Echo (ping) request id=0x0001, seq=17/4352, ttl=128 (reply in 64)
64	6.977744	172.20.10.2	172.20.10.7	ICMP	74	Echo (ping) reply id=0x0001, seq=17/4352, ttl=64 (request in 63)
65	7.960394	172.20.10.7	172.20.10.2	ICMP	74	Echo (ping) request id=0x0001, seq=18/4608, ttl=128 (no response found!)
66	7.960446	172.20.10.7	172.20.10.2	ICMP	74	Echo (ping) request id=0x0001, seq=18/4608, ttl=128 (reply in 67)
67	8.028355	172.20.10.2	172.20.10.7	ICMP	74	Echo (ping) reply id=0x0001, seq=18/4608, ttl=64 (request in 66)
68	8.978581	172.20.10.7	172.20.10.2	ICMP	74	Echo (ping) request id=0x0001, seq=19/4864, ttl=128 (no response found!)
69	8.978592	172.20.10.7	172.20.10.2	ICMP	74	Echo (ping) request id=0x0001, seq=19/4864, ttl=128 (reply in 70)
70	9.353684	172.20.10.2	172.20.10.7	ICMP	74	Echo (ping) reply id=0x0001, seq=19/4864, ttl=64 (request in 69)
74	9.989186	172.20.10.7	172.20.10.2	ICMP	74	Echo (ping) request id=0x0001, seq=20/5120, ttl=128 (no response found!)
75	9.989197	172.20.10.7	172.20.10.2	ICMP	74	Echo (ping) request id=0x0001, seq=20/5120, ttl=128 (reply in 78)
78	10.017512	172.20.10.2	172.20.10.7	ICMP	74	Echo (ping) reply id=0x0001, seq=20/5120, ttl=64 (request in 75)

Ilustración 1: Ping DNS Wireshark

En la Ilustración 2 se observa la captura de tráfico realizada en Wireshark durante la prueba de conectividad entre el cliente (172.20.10.7) y el servidor DNS (172.20.10.2). Mediante el filtro icmp, se identifican los paquetes Echo Request enviados desde el cliente al servidor y los correspondientes Echo Reply recibidos en respuesta. Cada request/reply se distingue por su identificador y número de secuencia, confirmando la comunicación bidireccional.

La información detallada de la captura permite corroborar los siguientes puntos:

- **Dirección IP de origen (request):** 172.20.10.7 (cliente).
- **Dirección IP de destino (request):** 172.20.10.2 (servidor DNS).
- **Dirección IP de origen (reply):** 172.20.10.2 (servidor DNS).
- **Dirección IP de destino (reply):** 172.20.10.7 (cliente).

Estos resultados coinciden con las estadísticas del comando ping, confirmando que la conectividad es exitosa y que no se registraron pérdidas de paquetes en la comunicación ICMP.

**1.2 Prueba ping FTP:** Realice pruebas de conectividad desde el Cliente al Servidor seleccionado para prestar el servicio de transferencia de archivos, para esta prueba utilice la dirección IP del servidor. Guarde en un archivo la captura del tráfico con el nombre Ping\_FTP\_IP.pcap.

```

Command Prompt - ftp 172.20.10.8
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Isabella Caputi> ping 172.20.10.8

Pinging 172.20.10.8 with 32 bytes of data:
Reply from 172.20.10.8: bytes=32 time<1ms TTL=64

Ping statistics for 172.20.10.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
  
```

Ilustración 2: Ping al servidor FTP con IP

El resultado del comando ping hacia el servidor FTP mediante la IP 172.20.10.8 evidencia conectividad correcta a nivel de red, ya que los cuatro paquetes enviados fueron recibidos sin pérdidas y con tiempos de respuesta inferiores a 1 ms, lo que confirma una latencia mínima propia de una red local. El valor de TTL=64 indica además que el destino corresponde a un sistema basado en Linux y que no existen múltiples saltos intermedios.

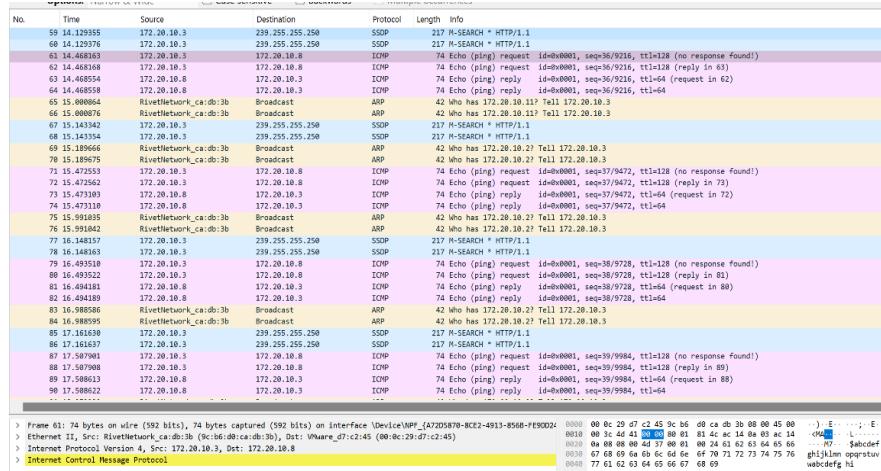


Ilustración 3: Ping FTP en Wireshark

Por otro lado, el análisis de la captura en Wireshark muestra que el cliente 172.20.10.3 envió múltiples solicitudes ICMP Echo (ping) al servidor 172.20.10.8 y todas fueron respondidas correctamente con ICMP Echo Reply, sin pérdida de paquetes. Cada solicitud y respuesta mantiene la secuencia esperada y un TTL=64, característico de sistemas Linux, lo que confirma que el servidor está activo y dentro de la misma red local. Por lo tanto, podemos concluir que estos resultados evidencian una conectividad estable y de baja latencia entre cliente y el servidor FTP.

**1.3 Abra cada uno de los archivos .pcap y en el cuadro de filtro en la interfaz de Wireshark escriba ICMP y haga click en el botón Apply; a continuación, identifique la siguiente información y consignela en una tabla en el documento de reporte:**

- Dirección IP de origen del paquete generado por el comando request

a. **Ping\_FTP\_IP.pcap**

The screenshot shows the Wireshark interface with a display filter set to "request". Two ICMP Echo (ping) request packets are visible. Both packets have a source IP of 172.20.10.3 and a destination IP of 172.20.10.8. The protocol is ICMP, and the length is 74 bytes. The first packet has an ID of 0x0001, sequence number 36, and TTL of 128. The second packet also has an ID of 0x0001, sequence number 36, and TTL of 128.

No.	Time	Source	Destination	Protocol	Length	Info
61	14.468163	172.20.10.3	172.20.10.8	ICMP	74	Echo (ping) request id=0x0001, seq=36/9216, ttl=128 (no response found!)
62	14.468168	172.20.10.3	172.20.10.8	ICMP	74	Echo (ping) request id=0x0001, seq=36/9216, ttl=128 (reply in 63)

Ilustración 5. Comando Request FTP

Como se observa en la Ilustración 5, el origen del paquete generado por el comando request es la IP 172.20.10.3.

b. **Ping\_DNS\_IP.pcap:**

The screenshot shows the Wireshark interface with a display filter set to "request". Six ICMP Echo (ping) request packets are visible. All packets have a source IP of 172.20.10.7 and a destination IP of 172.20.10.2. The protocol is ICMP, and the length is 74 bytes. The IDs range from 0x0001 to 0x0006, sequence numbers range from 5 to 1536, and TTL values range from 128 to 64.

No.	Time	Source	Destination	Protocol	Length	Info
186	12.539816	172.20.10.7	172.20.10.2	ICMP	74	Echo (ping) request id=0x0001, seq=5/1280, ttl=128 (no response found!)
187	12.539825	172.20.10.7	172.20.10.2	ICMP	74	Echo (ping) request id=0x0001, seq=5/1280, ttl=128 (reply in 188)
188	12.633899	172.20.10.2	172.20.10.7	ICMP	74	Echo (ping) reply id=0x0001, seq=5/1280, ttl=64 (request in 187)
255	13.548462	172.20.10.7	172.20.10.2	ICMP	74	Echo (ping) request id=0x0001, seq=6/1536, ttl=128 (no response found!)
256	13.548495	172.20.10.7	172.20.10.2	ICMP	74	Echo (ping) request id=0x0001, seq=6/1536, ttl=128 (reply in 257)
257	13.808350	172.20.10.2	172.20.10.7	ICMP	74	Echo (ping) reply id=0x0001, seq=6/1536, ttl=64 (request in 256)

Ilustración 6. Comando Request DNS

Como se observa en la Ilustración 6, el origen del paquete generado por el comando request es la IP 172.20.10.7.

- Dirección IP de destino del paquete generado por el comando request

- Ping\_FTP\_IP.pcap:** Como se observa en la Ilustración 5, el destino del paquete generado por el comando request es la IP 172.20.10.8 (la dirección del servidor FTP).
- Ping\_DNS\_IP.pcap:** Como se observa en la Ilustración 6, el destino del paquete generado por el comando request es la IP 172.20.10.2 (la dirección del servidor DNS).

- Dirección IP de origen del paquete generado por el comando reply

a. **Ping\_FTP\_IP.pcap**

The screenshot shows the Wireshark interface with a display filter set to "reply". Two ICMP Echo (ping) reply packets are visible. Both packets have a source IP of 172.20.10.3 and a destination IP of 172.20.10.8. The protocol is ICMP, and the length is 74 bytes. The IDs are 0x0001, sequence numbers are 36, and TTL values are 64.

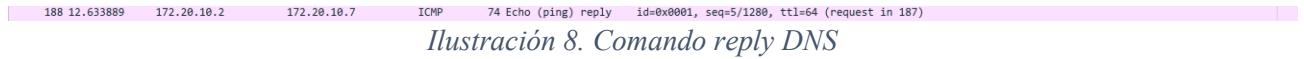
No.	Time	Source	Destination	Protocol	Length	Info
63	14.468554	172.20.10.8	172.20.10.3	ICMP	74	Echo (ping) reply id=0x0001, seq=36/9216, ttl=64 (request in 62)
64	14.468558	172.20.10.8	172.20.10.3	ICMP	74	Echo (ping) reply id=0x0001, seq=36/9216, ttl=64

Ilustración 7. Comando reply FTP

Como se observa en la Ilustración 7, el origen del paquete generado por el comando reply es la IP

172.20.10.8 (la dirección del servidor FTP).

**b. Ping\_DNS\_IP.pcap:**



Como se observa en la Ilustración 8, el origen del paquete generado por el comando reply es la IP 172.20.10.2 (la dirección del servidor DNS).

- Dirección IP de destino del paquete generado por el comando reply
  - a. **Ping\_FTP\_IP.pcap:** Como se observa en la Ilustración 7, el destino del paquete generado por el comando reply es la IP 172.20.10.3.
  - b. **Ping\_DNS\_IP.pcap:** Como se observa en la Ilustración 8, el destino del paquete generado por el comando reply es la IP 172.20.10.7.
- Dirección MAC del equipo Cliente

**a. Ping\_FTP\_IP.pcap**

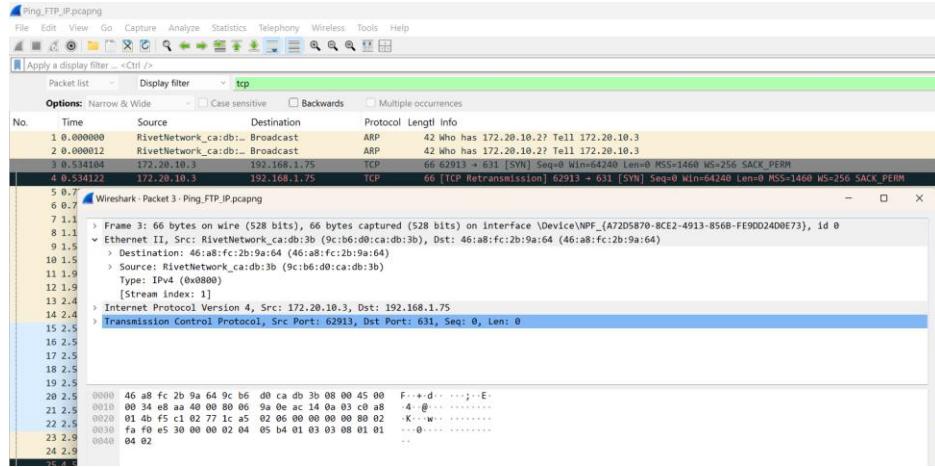


Ilustración 9. Direcciones MAC servidor FTP

Como se observa en la Ilustración 9, se hizo clic en el primer paquete enviado por protocolo TCP (línea 3). Al ver el detalle de Ethernet II la dirección MAC del equipo cliente (quien envía el paquete) es 9c:b6:d0:ca:db:3b.

**b. Ping\_DNS\_IP.pcap:**

Archivo / Prueba	Origen (Request)	Destino (Request)	Origen (Reply)	Destino (Reply)	MAC Cliente	MAC Servidor
Ping_FTP_IP.pcap	172.20.10.3	172.20.10.8	172.20.10.8	172.20.10.3	9c:b6:d0:ca:db:3b	46:a8:fc:2b:9a:64
Ping_DNS_IP.pcap	172.20.10.7	172.20.10.2	172.20.10.2	172.20.10.7	28:6b:35:d7:e0:24	a8:64:f1:bb:bb:15

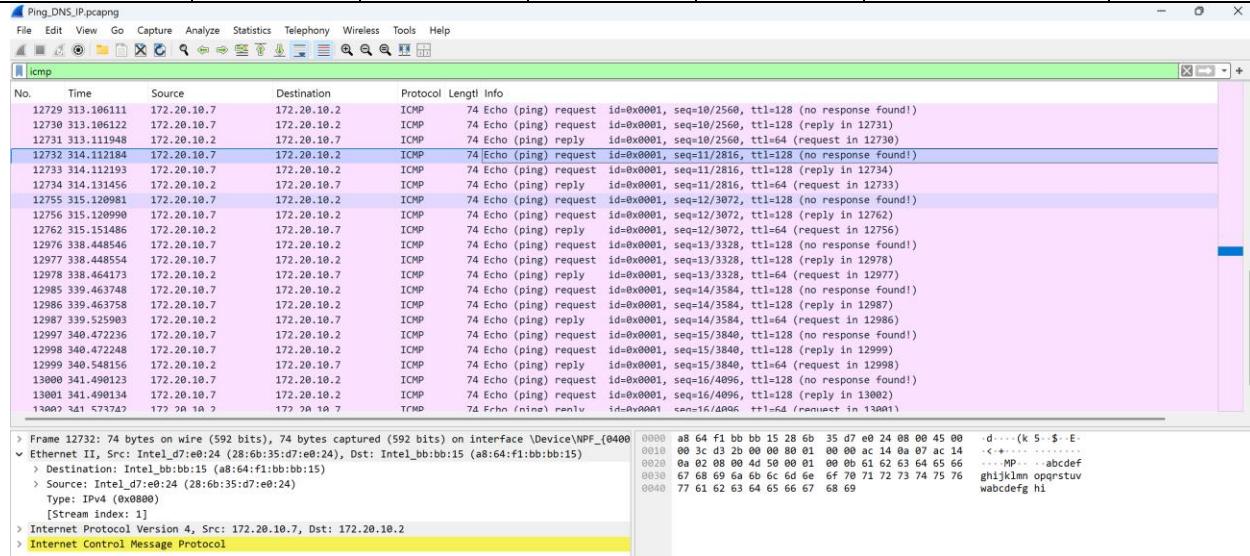


Ilustración 10. Direcciones MAC cliente DNS

Como se observa en la Ilustración 10, al ver el detalle de Ethernet II la dirección MAC del equipo cliente (quien envia el paquete) es 28:6b:35:d7:e0:24.

- Dirección MAC del equipo Servidor
  - Ping\_FTP\_IP.pcap:** Como se observa en la Ilustración 9, al ver el detalle de Ethernet, la dirección MAC del equipo servidor (quien recibe el paquete) es 46:a8:fc:2b:9a:64.
  - Ping\_DNS\_IP.pcap:** Como se observa en la Ilustración 9, al ver el detalle de Ethernet, la dirección MAC del equipo servidor (quien recibe el paquete) es a8:64:f1:bb:bb:15.

A continuación, se puede observar el resumen de todos los datos obtenidos de ambos archivos de Wireshark en la Tabla 1.

Tabla 1. Información Encontrada en los Archivos .pcap de Wireshark

## 2. Análisis de tráfico del Servicio DNS

**2.1 Realice pruebas de conectividad desde el Cliente al Servidor seleccionado como Servidor Web en la red utilizando su dirección IP. Guarde en un archivo la captura del tráfico con el nombre Ping\_WEB\_IP.pcap.**

Con el objetivo de validar la conectividad entre el cliente y el servidor designado para ofrecer el servicio Web dentro de la red, se ejecutó el comando ping empleando la dirección IP del servidor (172.20.10.5).

```
C:\Users\Sofia Toro>ping 172.20.10.5

Pinging 172.20.10.5 with 32 bytes of data:
Reply from 172.20.10.5: bytes=32 time=3ms TTL=64
Reply from 172.20.10.5: bytes=32 time<1ms TTL=64
Reply from 172.20.10.5: bytes=32 time<1ms TTL=64
Reply from 172.20.10.5: bytes=32 time=1ms TTL=64

Ping statistics for 172.20.10.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 3ms, Average = 1ms
```

Ilustración 11. Comando Ping a WEB con IP

Como podemos ver en la Ilustración anterior, en la prueba se enviaron 4 paquetes ICMP Echo Request y se recibieron 4 respuestas ICMP Echo Reply, sin pérdidas de paquetes (0% loss). Los tiempos de respuesta fueron notablemente bajos, con un mínimo de 0 ms, un máximo de 3 ms y un promedio de 1 ms, lo que evidencia una comunicación estable, directa y con latencia prácticamente nula entre cliente y servidor web.

Continuamos con las capturas de Wireshark:

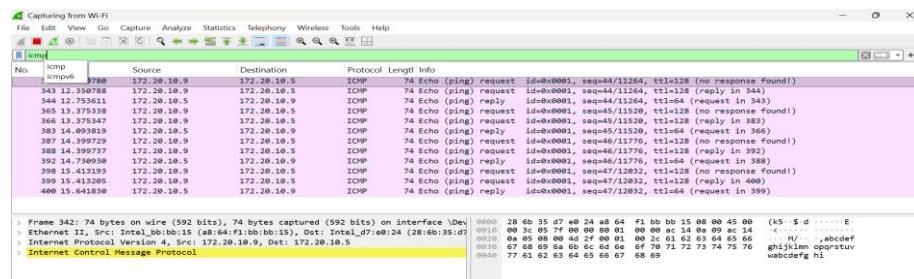


Ilustración 12. Ping a WEB en Wireshark

En la Ilustración 10 se presenta la captura de tráfico obtenida en Wireshark durante la prueba de conectividad entre el cliente (172.20.10.9) y el servidor Web (172.20.10.5).

Aplicando el filtro icmp, se identifican los paquetes correspondientes a la comunicación ICMP. Se observan los Echo Request enviados por el cliente al servidor y las respuestas Echo Reply que confirman la recepción.

Del análisis de la traza se evidencia:

- Dirección IP origen (request): 172.20.10.9 (cliente).
- Dirección IP destino (request): 172.20.10.5 (servidor web).
- Dirección IP origen (reply): 172.20.10.5 (servidor web).
- Dirección IP destino (reply): 172.20.10.9 (cliente).

Estos resultados concuerdan con los obtenidos en la consola mediante el comando ping, confirmando que la conectividad entre cliente y servidor web por IP es exitosa y estable.

## 2.2. En la consola de comandos del equipo Cliente digite el comando ipconfig/displaydns.

Con el fin de observar los registros DNS almacenados en el cliente, se utilizó el comando:

ipconfig /displaydns

```
C:\Users\Sofia Toro>ipconfig /displaydns
Windows IP Configuration

kv801.prod.do.dsp.mp.microsoft.com
-----
Record Name . . . . . : kv801.prod.do.dsp.mp.microsoft.com
Record Type . . . . . : 5
Time To Live . . . . . : 135777
Data Length . . . . . : 8
Section . . . . . : Answer
CNAME Record . . . . . : kv801.prod.do.dsp.mp.microsoft.com.edgekey.net

Record Name . . . . . : kv801.prod.do.dsp.mp.microsoft.com.edgekey.net
Record Type . . . . . : 5
Time To Live . . . . . : 135777
Data Length . . . . . : 8
Section . . . . . : Answer
CNAME Record . . . . . : e12437.d.akamaiedge.net

Record Name . . . . . : e12437.d.akamaiedge.net
Record Type . . . . . : 1
Time To Live . . . . . : 135777
Data Length . . . . . : 4
Section . . . . . : Answer
```

*Ilustración 13. Visualización de la caché DNS en el cliente*

En la Ilustración anterior se aprecia la salida generada por la consola, donde se muestran los registros resueltos recientemente. En este caso, el cliente contiene información asociada al dominio kv801.prod.do.dsp.mp.microsoft.com, el cual se encuentra configurado con un registro CNAME que apunta sucesivamente hacia otros dominios, tales como:

- kv801.prod.do.dsp.mp.microsoft.com.edgekey.net
- e12437.d.akamaiedge.net

Finalmente, este último dominio dispone de un registro de tipo A (Address Record) que corresponde a una dirección IPv4.

Estos resultados evidencian el funcionamiento del sistema DNS, en el cual un nombre de dominio puede estar asociado a múltiples alias antes de resolverse en una dirección IP concreta. La visualización de la caché DNS permite comprobar cómo el cliente almacena esta información para optimizar futuras consultas y reducir tiempos de resolución.

**2.3. Borre el registro caché del DNS. Para este fin en la consola de comando del equipo Cliente digite el comando ipconfig /flushdns. Verifique que la caché del DNS se vació usando nuevamente el comando ipconfig /displaydns.**

Con el propósito de limpiar los registros almacenados en la memoria caché del cliente, se ejecutó el comando:

ipconfig /flushdns

```
C:\Users\Sofia Toro>ipconfig /flushdns
Windows IP Configuration

Successfully flushed the DNS Resolver Cache.
```

*Ilustración 14. Limpieza de registros*

En la Ilustración anterior podemos ver como la consola confirmó la operación con el mensaje “*Successfully flushed the DNS Resolver Cache*”, indicando que el procedimiento se realizó correctamente.

Posteriormente, se verificó el estado de la caché mediante el comando:

ipconfig /displaydns

```

Successfully flushed the DNS Resolver Cache.

C:\Users\Sofia Toro>ipconfig /displaydns

Windows IP Configuration

  kubernetes.docker.internal
  -----
  No records of type AAAA

  Record Name . . . . . : kubernetes.docker.internal
  Record Type . . . . . : 1
  Time To Live . . . . . : 604800
  Data Length . . . . . : 4
  Section . . . . . : Answer
  A (Host) Record . . . . . : 127.0.0.1

```

Ilustración 15. Verificación de la caché DNS en el cliente

DNS en el cliente

En la Ilustración 14 se observa que los registros previos ya no aparecen y únicamente se mantiene una entrada local residual (kubernetes.docker.internal) con dirección IP 127.0.0.1, correspondiente a la configuración interna del sistema operativo.

Esto confirma que la caché DNS fue efectivamente vaciada y que el cliente se encuentra listo para realizar nuevas consultas DNS, las cuales serán resueltas directamente por el servidor DNS y no desde registros almacenados previamente.

**2.4. Realice nuevamente pruebas de conectividad desde el Cliente al Servidor seleccionado como Servidor Web en la red, pero esta vez utilizando su dirección URL (web.labredesZX.com). Guarde en un archivo la captura del tráfico con el nombre Ping\_WEB.pcap.**

Con el propósito de verificar la correcta resolución de nombres mediante el servicio DNS, se realizó una prueba de conectividad desde el cliente hacia el servidor Web utilizando su nombre de dominio web.labredes81.com, en lugar de la dirección IP.

```

C:\Users\Sofia Toro>ping web.labredes81.com

Pinging web.labredes81.com [172.20.10.5] with 32 bytes of data:
Reply from 172.20.10.5: bytes=32 time=1ms TTL=64
Reply from 172.20.10.5: bytes=32 time<1ms TTL=64
Reply from 172.20.10.5: bytes=32 time<1ms TTL=64
Reply from 172.20.10.5: bytes=32 time<1ms TTL=64

Ping statistics for 172.20.10.5:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

```

Ilustración 16. Prueba de conectividad al servidor Web mediante URL (ping en consola).

En la Ilustración se muestra el resultado del comando ping ejecutado en la consola. El dominio fue correctamente resuelto a la dirección IP 172.20.10.5 y se recibieron satisfactoriamente las 4 respuestas a los paquetes ICMP enviados. El tiempo de respuesta fue mínimo, con valores comprendidos entre 0 ms y 1 ms, reflejando una latencia muy baja y comunicación estable.

Continuamos con la captura de Wireshark:

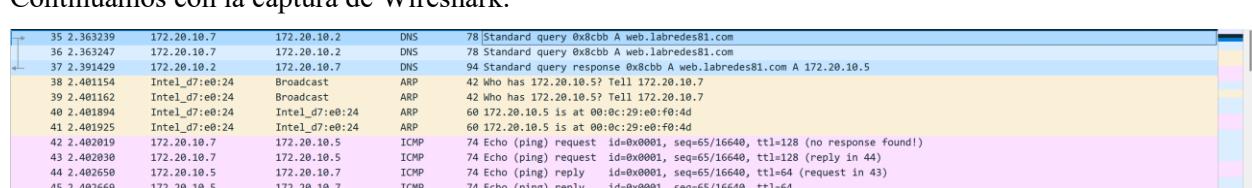


Ilustración 17. Captura en Wireshark mostrando la consulta/resuesta DNS y el intercambio ICMP

La Ilustración 16 corresponde a la captura en Wireshark durante la misma prueba. Allí se observa en primer lugar la consulta DNS (Standard query) enviada por el cliente (172.20.10.7) al servidor DNS (172.20.10.2), solicitando la resolución del nombre `web.labredes81.com`. Posteriormente, el servidor responde con la dirección IP 172.20.10.5, confirmando el funcionamiento del servicio DNS. A continuación, se visualizan los paquetes ICMP Echo Request y Echo Reply que confirman la conectividad entre cliente y servidor Web.

La captura completa fue almacenada en el archivo Ping\_WEB.pcap para su posterior análisis.

**2.5. Abra cada uno de los archivos. pcap y en el cuadro de filtro en la interfaz de Wireshark escriba DNS y haga click en el botón Apply; a continuación, identifique la siguiente información y consignela en el documento de reporte:**

- Identifique la información de la capa de aplicación que aparecen en los paquetes capturados que estén relacionados con el servicio DNS, en las pruebas realizadas.
- Identifique el protocolo de la capa de transporte generado por las peticiones al servidor DNS.
- Identifique los puertos utilizados por el servicio de DNS
- Complete la tabla generada en el punto anterior con los datos del Servidor W

Tras aplicar el filtro dns en Wireshark sobre los archivos Ping\_WEB\_IP.pcap y Ping\_WEB.pcap, se identificaron los paquetes relacionados con el servicio DNS, los cuales permiten detallar el proceso de resolución de nombres en la red.

Comenzamos con el Ping\_WEB.pcap

No.	Time	Source	Destination	Protocol	Length	Info
16	1.057649	172.20.10.7	172.20.10.2	DNS	98	Standard query 0x860f A self.events.data.microsoft.com
17	1.057673	172.20.10.7	172.20.10.2	DNS	98	Standard query 0x860f A self.events.data.microsoft.com
23	1.234874	172.20.10.2	172.20.10.7	DNS	98	Standard query response 0x860f Server failure A self.events.data.microsoft.com
26	1.299243	172.20.10.7	172.20.10.2	DNS	86	Standard query 0x73d1 A uniandes-my.sharepoint.com
27	1.299253	172.20.10.7	172.20.10.2	DNS	86	Standard query 0x73d1 A uniandes-my.sharepoint.com
32	1.385596	172.20.10.2	172.20.10.7	DNS	86	Standard query response 0x73d1 Server failure A uniandes-my.sharepoint.com
35	2.363239	172.20.10.7	172.20.10.2	DNS	78	Standard query 0x8ccb A web.labredes81.com
36	2.363247	172.20.10.7	172.20.10.2	DNS	78	[Standard query 0x8ccb A web.labredes81.com]
37	2.391429	172.20.10.2	172.20.10.7	DNS	94	Standard query response 0x8ccb A web.labredes81.com A 172.20.10.5
46	2.891731	172.20.10.7	172.20.10.2	DNS	89	Standard query 0xcc24 A collabrtc.officeapps.live.com
47	2.891744	172.20.10.7	172.20.10.2	DNS	89	Standard query 0xcc24 A collabrtc.officeapps.live.com
50	3.081974	172.20.10.2	172.20.10.7	DNS	89	Standard query response 0xcc24 Server failure A collabrtc.officeapps.live.com
51	3.245448	172.20.10.7	172.20.10.2	DNS	92	Standard query 0x022c A mobile.events.data.microsoft.com
52	3.245457	172.20.10.7	172.20.10.2	DNS	92	Standard query 0x022c A mobile.events.data.microsoft.com
53	3.149598	172.20.10.2	172.20.10.7	DNS	92	Standard query response 0x022c Server failure A mobile.events.data.microsoft.com
60	4.148345	172.20.10.7	172.20.10.2	DNS	76	Standard query 0x1c12 A b.slack-edge.com
61	4.148357	172.20.10.7	172.20.10.2	DNS	76	Standard query 0x1c12 A b.slack-edge.com
62	4.148429	172.20.10.7	172.20.10.2	DNS	76	Standard query 0x2a10 A a.slack-edge.com
63	4.148440	172.20.10.7	172.20.10.2	DNS	76	Standard query 0x2a10 A a.slack-edge.com
64	4.281281	172.20.10.2	172.20.10.7	DNS	76	Standard query response 0x2a10 Server failure A a.slack-edge.com
65	4.363168	172.20.10.7	172.20.10.2	DNS	76	Standard query response 0x1c17 Server failure A b.slack-edge.com

> Frame 36: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) interface \Device\NPF\_{04082114-0000-4000-8000-000000000000  
> Ethernet II, Src: Intel\_d7:e0:24 (28:6b:35:d7:e0:24), Dst: Intel\_bb:bb:15 (a8:64:f1:bb:bb:15)  
> Internet Protocol Version 4, Src: 172.20.10.7, Dst: 172.20.10.2  
> User Datagram Protocol, Src Port: 53281, Dst Port: 53  
> Domain Name System (query)

Ilustración 18. Captura en Wireshark mostrando query `web.labredes81.com`

En la Ilustración anterior podemos evidenciar como se realizar el query al DNS se `web.labredes81.com`.

A continuación, se puede ver como se responde a ese query:

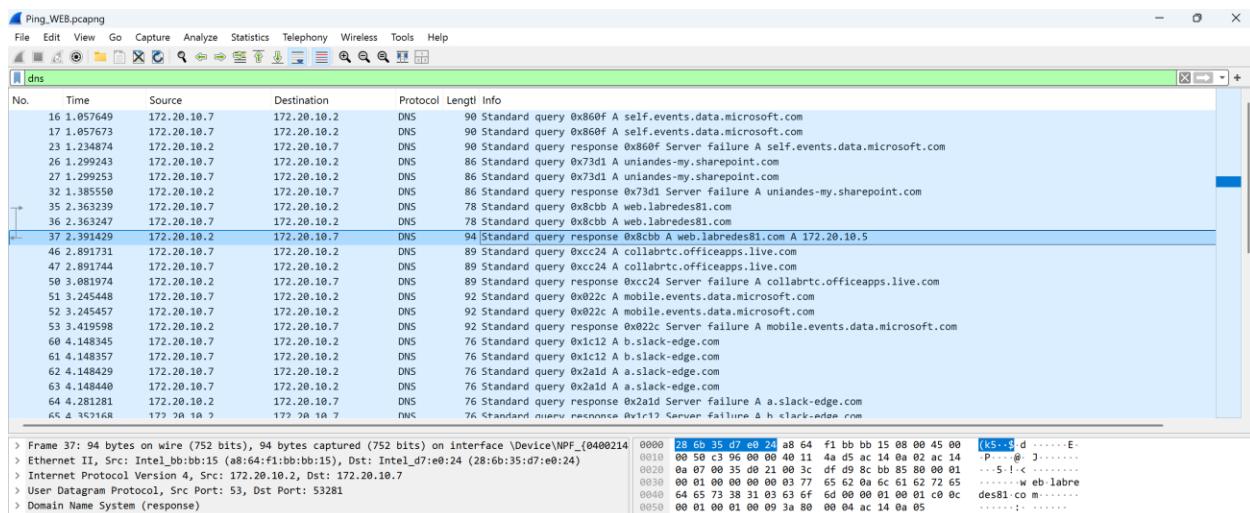


Ilustración 19. Captura en Wireshark mostrando response web.labredes81.com

En la Ilustración anterior podemos ver como se resuelve la URL a la IP correspondiente a el servidor Web.

### Información observada en la capa de aplicación

En la columna **Info** de los paquetes filtrados se muestran:

- Standard query A web.labredes81.com** la cual es la petición realizada por el cliente al servidor DNS solicitando la IP asociada al dominio.
- Standard query response A web.labredes81.com A 172.20.10.5** la cual es la respuesta del servidor DNS, devolviendo la dirección IP que corresponde al dominio consultado.

### Protocolo de la capa de transporte

Las peticiones y respuestas DNS utilizan el protocolo UDP (User Datagram Protocol), propio del servicio DNS para consultas rápidas y ligeras. Esto se evidencia en la parte inferior izquierda de la Ilustración 17 y 18.

### Puertos utilizados

- Puerto origen (cliente):** 53281
- Puerto destino (servidor):** 53, reservado para el servicio DNS.

Prueba	Dominio consultado	IP resuelta	IP Cliente (origen)	IP Servidor DNS (destino)	Protocolo transporte	Puerto destino	Puerto origen
Ping Web por URL	web.labredes81.com	172.20.10.5	172.20.10.7	172.20.10.2	UDP	53	53281

### **3. Análisis de tráfico del Servicio FTP**

El protocolo de transferencia de archivos (FTP) se utiliza para transferir archivos desde un dispositivo de red hasta otro. Verifique el correcto funcionamiento para descargar y cargar archivos del servidor antes de iniciar el trabajo propuesto.

#### **3.1 Borre el registro caché del DNS.**

Como se realizó anteriormente en el análisis de tráfico del DNS, se ejecutó en la consola del cliente el comando: ipconfig /flushdns para borrar el registro de cache del DNS.

The image shows two separate command-line windows from a Windows operating system. The top window is titled 'Windows IP Configuration' and displays the result of the command 'ipconfig /displaydns'. It shows a section for 'kubernetes.docker.internal' with the message 'No records of type AAAA'. The bottom window is also titled 'Windows IP Configuration' and displays the result of the command 'ipconfig /flushdns'. It shows a message 'Successfully flushed the DNS Resolver Cache.' followed by detailed information about a flushed entry for 'kubernetes.docker.internal': Record Name . . . . : kubernetes.docker.internal, Record Type . . . . : 1, Time To Live . . . . : 565974, Data Length . . . . : 4, Section . . . . : Answer, A (Host) Record . . . : 127.0.0.1.

*Ilustración 20. Flush del Registro cache del DNS*

*Ilustración 21. Flush del Registro cache del DNS*

Como se observa en la ilustración 20 y 21, se observa, como en el punto anterior, que los registros previos ya no aparecen y que la caché DNS fue efectivamente vaciada. Por lo tanto, las consultas del cliente serán resueltas directamente por el servidor DNS y no desde registros almacenados previamente.

#### **3.2 En la maquina Cliente, acceda al programa cliente FTP y conéctese al servidor FTP, utilice la dirección IP del servidor o su URL ([ftp.labredesZX.com](http://ftp.labredesZX.com)). Recuerde que para acceder al servicio debe autenticarse con un usuario valido.**

En la ilustración 22, podemos observar la conexión al servidor FTP utilizando su IP. Luego, se autentica como el usuario1.

The image shows a terminal window with the following text output:

```
PS C:\Users\Isabella Caputi> ftp 172.20.10.8
Connected to 172.20.10.8.
220 ProFTPD Server (Debian) [::ffff:172.20.10.8]
200 UTF8 set to on
User (172.20.10.8:(none)): usuario1
331 Password required for usuario1
Password:
230 User usuario1 logged in
ftp> ls
200 PORT command successful
150 Opening ASCII mode data connection for file list
usuario1
226 Transfer complete
ftp: 13 bytes received in 0.00Seconds 6.50Kbytes/sec.
ftp>
```

*Ilustración 22. Acceso al programa cliente FTP con IP*

#### **3.3 Abra cada uno de los archivos. pcap y realice primero un filtro en la interfaz de Wireshark para el protocolo FTP y haga click en el botón Apply; a continuación, identifique la siguiente información y consíguela en el documento de reporte:**

- Identifique la información de la capa de aplicación que aparecen en los paquetes capturados que estén relacionados con el servicio de transferencia de archivo, en las pruebas realizadas.**

### a. FTP\_download.pcap

No.	Tim	Source	Destination	Protocol	Length	Info
194	40.869379	172.20.10.3	172.20.10.8	FTP	80	Request: PORT 172,20,10,3,245,191
196	40.871220	172.20.10.8	172.20.10.3	FTP	83	Response: 200 PORT command successful
198	40.882384	172.20.10.3	172.20.10.8	FTP	68	Request: RETR c21.sql
206	40.884033	172.20.10.8	172.20.10.3	FTP	106	Response: 150 Opening ASCII mode data connection for c21.sql
216	40.901971	172.20.10.8	172.20.10.3	FTP	77	Response: 226 Transfer complete

Ilustración 23. Información Capa de Aplicación Descarga

En la captura mostrada se evidencia la operación de descarga de un archivo desde el servidor FTP hacia el cliente. Primero, el cliente establece el canal de datos mediante el comando PORT, recibiendo la confirmación del servidor con el código 200. Posteriormente, se observa el uso del comando RETR c21.sql, el cual indica la solicitud de obtener el archivo desde el servidor. Como respuesta, el servidor emite el código 150, señalando la apertura de la conexión de datos en modo ASCII para transferir el archivo solicitado. Finalmente, el mensaje 226 Transfer complete confirma que la descarga del archivo c21.sql se completó de manera exitosa.

### b. FTP\_upload.pcap

4458 601.789729	172.20.10.8	172.20.10.3	FTP	77	Response: 226 Transfer complete
4571 618.765451	172.20.10.8	172.20.10.3	FTP	104	Response: 220 ProFTPD Server (Debian) [::ffff:172.20.10.8]
4573 618.765783	172.20.10.3	172.20.10.8	FTP	64	Request: AUTH TLS
4577 618.766687	172.20.10.8	172.20.10.3	FTP	79	Response: 500 AUTH not understood
4579 618.766923	172.20.10.3	172.20.10.8	FTP	64	Request: AUTH SSL
4581 618.767559	172.20.10.8	172.20.10.3	FTP	79	Response: 500 AUTH not understood
4583 618.771531	172.20.10.3	172.20.10.8	FTP	69	Request: USER usuariol
4585 618.773073	172.20.10.8	172.20.10.3	FTP	99	Response: 333 Password required for usuariol
4587 618.773348	172.20.10.3	172.20.10.8	FTP	62	Request: PASS c21.sql
4591 618.834345	172.20.10.8	172.20.10.3	FTP	83	Response: 230 User usuariol logged in
4593 618.834594	172.20.10.3	172.20.10.8	FTP	70	Request: CLNT FileZilla
4597 618.835279	172.20.10.8	172.20.10.3	FTP	62	Response: 200 OK
4599 618.835424	172.20.10.3	172.20.10.8	FTP	68	Request: OPTS UTF8 ON
4601 618.835898	172.20.10.8	172.20.10.3	FTP	74	Response: 205 UTF8 set to on
4603 618.838187	172.20.10.3	172.20.10.8	FTP	69	Request: CWD /usuariol
4605 618.840063	172.20.10.8	172.20.10.3	FTP	82	Response: 250 Cmd command successful
4607 618.840987	172.20.10.3	172.20.10.8	FTP	62	Request: TYPE A
4609 618.841677	172.20.10.8	172.20.10.3	FTP	73	Response: 200 Type set to A
4611 618.842001	172.20.10.3	172.20.10.8	FTP	60	Request: PASS
4613 618.842518	172.20.10.8	172.20.10.3	FTP	103	Response: 237 Entering Passive Mode (172,20,10,8,141,67).
4615 618.842809	172.20.10.3	172.20.10.8	FTP	68	Request: STOR c21.sql
4625 618.843884	172.20.10.8	172.20.10.3	FTP	106	Response: 150 Opening ASCII mode data connection for c21.sql
4631 618.844465	172.20.10.8	172.20.10.3	FTP	77	Response: 226 Transfer complete
4647 620.549683	172.20.10.3	172.20.10.8	FTP	60	Request: PASV
4649 620.550715	172.20.10.8	172.20.10.3	FTP	104	Response: 227 Entering Passive Mode (172,20,10,8,146,137).
4651 620.550991	172.20.10.3	172.20.10.8	FTP	60	Request: MLSD
4659 620.551826	172.20.10.8	172.20.10.3	FTP	104	Response: 150 Opening BINARY mode data connection for MLSD
4669 620.552858	172.20.10.8	172.20.10.3	FTP	77	Response: 226 Transfer complete

Ilustración 24. Información Capa de Aplicación Carga

En la captura de la ilustración 24 se muestra la carga de un archivo mediante FTP se evidencia la autenticación del cliente mediante los comandos USER y PASS, seguida de la aceptación del servidor con el código 230 (inicio de sesión exitoso). El cliente ejecutó el comando STOR c21.sql, al cual el servidor respondió con el código 150 (apertura de conexión de datos) y finalmente con el código 226, que confirma la transferencia exitosa del archivo. Esto demuestra el correcto funcionamiento del proceso de carga en el servicio FTP.

- Identifique el protocolo de la capa de transporte generado por las peticiones al servidor FTP.

### a. FTP\_download.pcap

No.	Time	Source	Destination	Protocol	Length	Info
191	40.733628	fe80::53d:3896:ef7b.. ff02::fb		MDNS	95	Standard query 0x0000 PTR _ipp._tcp.local, "QM" question
192	40.839807	RivetNetwork_ca:db:..	Broadcast	ARP	42	Who has 172.20.10.2? Tell 172.20.10.3
193	40.839821	RivetNetwork_ca:db:..	Broadcast	ARP	42	Who has 172.20.10.2? Tell 172.20.10.3
194	40.869378	172.20.10.3	172.20.10.8	FTP	80	Request: PORT 172,20,10,3,245,191
195	40.869397	172.20.10.3	172.20.10.8	TCP	83	[TCP Retransmission] 62874 + 21 [PSH, ACK] Seq=1 Ack=1 Win=7711 Len=26
196	40.871220	172.20.10.3	172.20.10.8	FTP	83	Response: 200 PORT command successful
197	40.871233	172.20.10.3	172.20.10.8	TCP	68	[TCP Retransmission] 21 + 62874 [PSH, ACK] Seq=1 Ack=27 Win=502 Len=29
198	40.882384	172.20.10.3	172.20.10.8	FTP	68	Request: RETR c21.sql
199	40.882421	172.20.10.3	172.20.10.8	TCP	68	[TCP Retransmission] 62874 + 21 [PSH, ACK] Seq=27 Ack=30 Win=7682 Len=14
200	40.883322	172.20.10.3	172.20.10.8	TCP	74	74 [TCP Retransmission] 41773 + 62911 [SYN] Seq=0 Win=64240 Len=0 MSS=1468 SACK_PERM Tsvl=2006485967 TSecr=0 WS=128
201	40.883330	172.20.10.3	172.20.10.8	TCP	74	[TCP Retransmission] 41773 + 62911 [SYN] Seq=0 Win=64240 Len=0 MSS=1468 SACK_PERM Tsvl=2006485967 TSecr=0 WS=128
202	40.883464	172.20.10.3	172.20.10.8	TCP	74	62911 + 41773 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1468 WS=256 SACK_PERM Tsvl=90100878 TSecr=2006485967
203	40.883472	172.20.10.3	172.20.10.8	TCP	74	[TCP Retransmission] 62911 + 41773 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1468 WS=256 SACK_PERM Tsvl=90100878
204	40.883715	172.20.10.3	172.20.10.8	TCP	68	614773 + 62911 [ACK] Seq=1 Ack=1 Win=64256 Len=0 Tsvl=2006485967 TSecr=90100878
205	40.883718	172.20.10.3	172.20.10.8	TCP	68	[TCP Dup ACK 204 1] 41773 + 62911 [ACK] Seq=0 Ack=1 Win=64256 Len=0 Tsvl=2006485967 TSecr=90100878
206	40.884033	172.20.10.3	172.20.10.8	FTP	108	Response: 150 Opening ASCII mode data connection for c21.sql
207	40.884037	172.20.10.3	172.20.10.8	TCP	108	[TCP Retransmission] 21 + 62874 [PSH, ACK] Seq=30 Ack=41 Win=502 Len=52
208	40.884312	172.20.10.3	172.20.10.8	TCP	68	614773 + 62911 [FIN, ACK] Seq=1 Ack=1 Win=64256 Len=0 Tsvl=2006485968 TSecr=9010078
209	40.884317	172.20.10.3	172.20.10.8	TCP	66	[TCP Retransmission] 41773 + 62911 [FIN, ACK] Seq=1 Ack=1 Win=64256 Len=0 Tsvl=2006485968 TSecr=9010078
210	40.884355	172.20.10.3	172.20.10.8	TCP	66	62911 + 41773 [ACK] Seq=1 Ack=2 Win=1049600 Len=0 Tsvl=9010079 TSecr=2006485968
211	40.884361	172.20.10.3	172.20.10.8	TCP	66	[TCP Dup ACK 210 1] 62911 + 41773 [ACK] Seq=1 Ack=2 Win=1049600 Len=0 Tsvl=9010079 TSecr=2006485968
212	40.900330	172.20.10.3	172.20.10.8	TCP	66	62911 + 41773 [FIN, ACK] Seq=0 Ack=2 Win=1049600 Len=0 Tsvl=9010095 TSecr=2006485968
213	40.900339	172.20.10.3	172.20.10.8	TCP	66	[TCP Retransmission] 62911 + 41773 [FIN, ACK] Seq=1 Ack=2 Win=1049600 Len=0 Tsvl=9010095 TSecr=2006485968
214	40.900860	172.20.10.3	172.20.10.8	TCP	66	614773 + 62911 [ACK] Seq=0 Ack=2 Win=64256 Len=0 Tsvl=2006485985 TSecr=9010095
215	40.900869	172.20.10.3	172.20.10.8	TCP	66	[TCP Dup ACK 214 1] 41773 + 62911 [ACK] Seq=2 Ack=2 Win=64256 Len=0 Tsvl=2006485985 TSecr=9010095
216	40.901971	172.20.10.8	172.20.10.3	FTP	77	Response: 226 Transfer complete
217	40.902021	172.20.10.8	172.20.10.3	TCP	77	[TCP Retransmission] 21 + 62874 [PSH, ACK] Seq=82 Ack=41 Win=502 Len=23
218	40.902095	172.20.10.3	172.20.10.8	TCP	54	62874 + 21 [ACK] Seq=41 Ack=105 Win=7607 Len=0
219	40.902119	172.20.10.3	172.20.10.8	TCP	54	[TCP Dup ACK 218 1] 62874 + 21 [ACK] Seq=41 Ack=105 Win=7607 Len=0
220	41.844066	RivetNetwork_ca:db:..	Broadcast	ARP	42	Who has 172.20.10.2? Tell 172.20.10.3
221	41.844078	RivetNetwork_ca:db:..	Broadcast	ARP	42	Who has 172.20.10.2? Tell 172.20.10.3

Ilustración 25. Protocolo de Transporte Descarga

En la captura de Wireshark se observa que las peticiones realizadas al servidor FTP se transportan mediante el protocolo TCP, el cual aparece en los paquetes asociados al servicio. Este protocolo garantiza la comunicación confiable entre cliente y servidor, utilizando el puerto 21 para el canal de control y puertos dinámicos para la transferencia de datos, lo que confirma que el servicio FTP depende de TCP en la capa de transporte.

## b. FTP\_upload.pcapng

No.	Time	Source	Destination	Protocol	Length	Info
4423	601.784609	172.20.10.3	172.20.10.8	TCP	82	[TCP Retransmission] 21 + 62883 [PSH, ACK] Seq=228 Ack=94 Win=64256 Len=28
4424	601.784896	172.20.10.3	172.20.10.8	FTP	62	Request: TYPE I
4425	601.784903	172.20.10.3	172.20.10.8	TCP	62	[TCP Retransmission] 62883 + 21 [PSH, ACK] Seq=94 Ack=256 Win=1049344 Len=8
4426	601.785267	172.20.10.8	172.20.10.3	FTP	73	73 Response: 200 Type set to I
4427	601.785634	172.20.10.3	172.20.10.8	TCP	73	[TCP Retransmission] 21 + 62883 [PSH, ACK] Seq=256 Ack=102 Win=64256 Len=19
4428	601.785805	172.20.10.3	172.20.10.8	FTP	60	60 Request: PASV
4429	601.785810	172.20.10.3	172.20.10.8	TCP	60	[TCP Retransmission] 62883 + 21 [PSH, ACK] Seq=102 Ack=275 Win=1049344 Len=6
4430	601.786531	172.20.10.8	172.20.10.3	FTP	102	102 Response: 227 Entering Passive Mode (172,20,10,8,166,5).
4431	601.786534	172.20.10.3	172.20.10.8	TCP	102	[TCP Retransmission] 21 + 62883 [PSH, ACK] Seq=275 Ack=108 Win=64256 Len=48
4432	601.786865	172.20.10.3	172.20.10.8	FTP	60	60 Request: MLSD
4433	601.786872	172.20.10.3	172.20.10.8	TCP	60	[TCP Retransmission] 62883 + 21 [PSH, ACK] Seq=108 Ack=323 Win=1049344 Len=6
4434	601.787451	172.20.10.3	172.20.10.8	TCP	66	62884 + 42501 [SYN] Seq=0 Win=65535 Len=0 MSS=1468 WS=128 SACK_PERM
4435	601.787460	172.20.10.3	172.20.10.8	TCP	66	[TCP Retransmission] 62884 + 42501 [SYN] Seq=0 Win=65535 MSS=1460 WS=128 SACK_PERM
4436	601.787808	172.20.10.8	172.20.10.3	TCP	66	642501 + 62884 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
4437	601.787812	172.20.10.8	172.20.10.3	TCP	66	[TCP Retransmission] 42501 + 62884 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
4438	601.787901	172.20.10.3	172.20.10.8	TCP	54	62884 + 42501 [ACK] Seq=1 Ack=1 Win=4194304 Len=0
4439	601.787907	172.20.10.3	172.20.10.8	TCP	54	[TCP Dup ACK 4438 1] 62884 + 42501 [ACK] Seq=1 Ack=1 Win=4194304 Len=0
4440	601.788413	172.20.10.8	172.20.10.3	FTP	104	104 Response: 150 Opening BINARY mode data connection for MLSD
4441	601.788418	172.20.10.8	172.20.10.3	TCP	104	[TCP Retransmission] 21 + 62883 [PSH, ACK] Seq=323 Ack=114 Win=64256 Len=50
4442	601.789053	172.20.10.8	172.20.10.3	FTP-DATA	358	358 FTP Data: 304 bytes (PASV) (MLSD)
4443	601.789066	172.20.10.8	172.20.10.3	TCP	54	[TCP Retransmission] 42501 + 62884 [FIN, PSH, ACK] Seq=1 Ack=1 Win=64256 Len=304
4444	601.789127	172.20.10.3	172.20.10.8	TCP	54	54 [TCP Dup ACK 42501 1] 62884 + 42501 [ACK] Seq=1 Ack=1 Win=4193920 Len=0
4445	601.789133	172.20.10.3	172.20.10.8	TCP	54	[TCP Dup ACK 4444 1] 62884 + 42501 [ACK] Seq=1 Ack=1 Win=4193920 Len=0
4446	601.789261	172.20.10.3	172.20.10.8	TCP	54	54 [TCP Dup ACK 42501 1] 62884 + 42501 [ACK] Seq=1 Ack=306 Win=4193920 Len=0
4447	601.789267	172.20.10.3	172.20.10.8	TCP	54	[TCP Retransmission] 62884 + 42501 [FIN, ACK] Seq=1 Ack=306 Win=4193920 Len=0
4448	601.789507	172.20.10.3	172.20.10.8	TCP	60	60 42501 + 62884 [ACK] Seq=306 Ack=2 Win=64256 Len=0
4449	601.789508	172.20.10.8	172.20.10.3	TCP	60	[TCP Dup ACK 4448 1] 42501 + 62884 [ACK] Seq=306 Ack=2 Win=64256 Len=0
4450	601.789729	172.20.10.8	172.20.10.3	FTP	77	77 Response: 226 Transfer complete
4451	601.789734	172.20.10.8	172.20.10.3	TCP	77	[TCP Retransmission] 21 + 62883 [PSH, ACK] Seq=373 Ack=114 Win=64256 Len=23
4452	601.789763	172.20.10.3	172.20.10.8	TCP	54	54 [TCP Dup ACK 4452 1] 62883 + 21 [ACK] Seq=114 Ack=396 Win=1049344 Len=0
4453	601.789769	172.20.10.3	172.20.10.8	TCP	54	[TCP Dup ACK 4452 1] 62883 + 21 [ACK] Seq=114 Ack=396 Win=1049344 Len=0

Ilustración 26. Protocolo de Transporte Carga

De nuevo, en la ilustración 26 se observa el caso de la carga de archivos, en el cual observamos que las peticiones realizadas al servidor FTP se transportan mediante el protocolo TCP. Este protocolo garantiza la comunicación confiable entre cliente y servidor, utilizando el puerto 21 para el canal de control y

puertos dinámicos para la transferencia de datos, lo que confirma de nuevo que el servicio FTP depende de TCP en la capa de transporte.

- **Identifique los puertos utilizados por el servicio de FTP**

- a. **FTP\_download.pcap**

En la captura de Wireshark de la ilustración 25 se identificó que el servicio FTP utiliza el puerto 21/TCP como canal de control, a través del cual se transmiten los comandos y respuestas entre el cliente y el servidor (PORT, RETR). Asimismo, para la transferencia de archivos se emplean puertos dinámicos o efímeros (por ejemplo, 41773 o 62911 observados en la traza de Wireshark), que son asignados de manera temporal con el fin de establecer el canal de datos. De esta forma, se confirma que el protocolo FTP opera utilizando un puerto fijo para el control y puertos variables para la transmisión de la información, asegurando así la separación funcional entre el envío de instrucciones y la transferencia de archivos.

- b. **FTP\_upload.pcap**

En la captura correspondiente a la operación de carga de archivos mediante FTP se identificó el uso del puerto 21 en el servidor para el canal de control al igual que en el de descarga, a través del cual se intercambiaron los comandos de la sesión. Para la transferencia de datos se utilizaron puertos dinámicos. Se identifican el puerto 62884 en el cliente y el puerto 42501 en el servidor, evidenciando así los puertos implicados en la fase de carga del archivo.

## 4. Análisis de tráfico del Servicio Web

El servidor web es uno de los servicios de red más populares que utiliza un modelo cliente/servidor. Se utilizan principalmente los protocolos HTTP y HTTPS. Verifique que tiene conectividad con el servidor y que su servicio web funciona correctamente accediendo desde el navegador de la máquina Cliente.

### 4.1 Borre el registro caché del DNS.

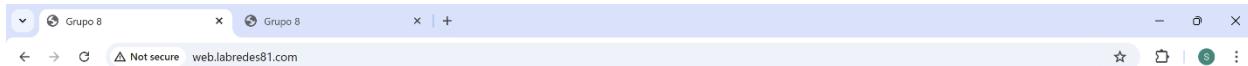
Para garantizar que las pruebas de resolución de nombres se realizaran sin información almacenada previamente en la caché local, se ejecutó en la consola del cliente el comando: ipconfig /flushdns

```
C:\Users\Sofia Toro>ipconfig /flushdns
Windows IP Configuration
Successfully flushed the DNS Resolver Cache.
```

Ilustración 27 Borrar registro cache DNS

La Ilustración anterior muestra el resultado del procedimiento, donde el sistema confirma: “*Successfully flushed the DNS Resolver Cache*”. Esto indica que la memoria caché de DNS fue vaciada correctamente y que las siguientes consultas se realizarán directamente contra el servidor DNS, evitando resultados almacenados con anterioridad.

### 4.2 Utilizando el navegador de la máquina cliente conéctese al sitio web configurado en su servidor utilizando el protocolo HTTP. Guarde en un archivo la captura del tráfico generado con el nombre **HTTP\_view.pcap**.

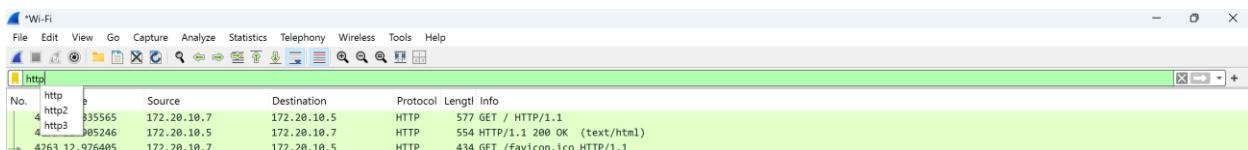


## Hello World - Somos el Grupo 8.

Esta es nuestra primera página. Servidor Web funcionado

*Ilustración 28 HTTP pagina web.*

Una vez borrada la caché de DNS, se procedió a acceder desde el navegador del equipo cliente al sitio web configurado en el servidor, utilizando el protocolo **HTTP**. Durante esta acción, se registró el tráfico en Wireshark y se guardó en el archivo **HTTP\_view.pcap**.



*Ilustración 29 Captura Wireshark del intercambio de mensajes HTTP (GET, 200 OK, 404 Not Found).*

En la Ilustración se observa la petición HTTP enviada desde el cliente (172.20.10.7) hacia el servidor web (172.20.10.5) con el método GET / HTTP/1.1, solicitando la página principal. El servidor responde con un código 200 OK, indicando que la solicitud fue procesada correctamente y que el recurso se devolvió en formato text/html.

4159	172.20.10.2	172.20.10.7	TCP	60	53 → 60459 [FIN, ACK] Seq=53 ACK=56 Win=64/68 Len=0
4160	172.20.10.2	172.20.10.7	DNS	108	Standard query response 0xe0fe4 A web.labredes81.com A 172.20.10.5
4161	172.20.10.7	172.20.10.2	TCP	54	60459 → 53 [ACK] Seq=56 Ack=56 Win=65280 Len=0
4162	172.20.10.7	172.20.10.2	TCP	54	[TCP Dup ACK 4161#1] 60459 → 53 [ACK] Seq=56 Ack=56 Win=65280 Len=0
4163	172.20.10.2	172.20.10.7	DNS	134	Standard query response 0x9702 HTTPS web.labredes81.com SOA labredes81.com
4164	172.20.10.7	172.20.10.2	TCP	54	59335 → 53 [FIN, ACK] Seq=59 Ack=55 Win=65280 Len=0
4165	172.20.10.7	172.20.10.2	TCP	54	[TCP Retransmission] 59335 → 53 [FIN, ACK] Seq=59 Ack=55 Win=65280 Len=0
4166	172.20.10.7	172.20.10.5	TCP	66	62874 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM

*Ilustración 30.Resolución DNS y establecimiento de conexión TCP para la sesión HTTP.*

En la Ilustración anterior, se evidencian las etapas previas a la comunicación HTTP:

- Primero, la consulta y respuesta DNS donde se resolvió el dominio *web.labredes81.com* a la dirección IP 172.20.10.5.
- Luego, el establecimiento de la sesión TCP a través de un three-way handshake entre cliente y servidor en el puerto 80, reservado para HTTP.

Estos resultados confirman que el servicio web está operativo y que el cliente puede conectarse satisfactoriamente al servidor utilizando el protocolo HTTP.

### 4.3 Abra el archivo **HTTP\_view.pcap** y realice primero un filtro en la interfaz de Wireshark para el protocolo **HTTP** y haga click en el botón **Apply**; a continuación, identifique la siguiente información y consíguela en el documento de reporte:

- Identifique la información de la capa de aplicación que aparecen en los paquetes capturados que estén relacionados con el servicio web.
- Identifique el protocolo de la capa de transporte generado por las peticiones al servidor web.
- Identifique los puertos utilizados por el servicio web

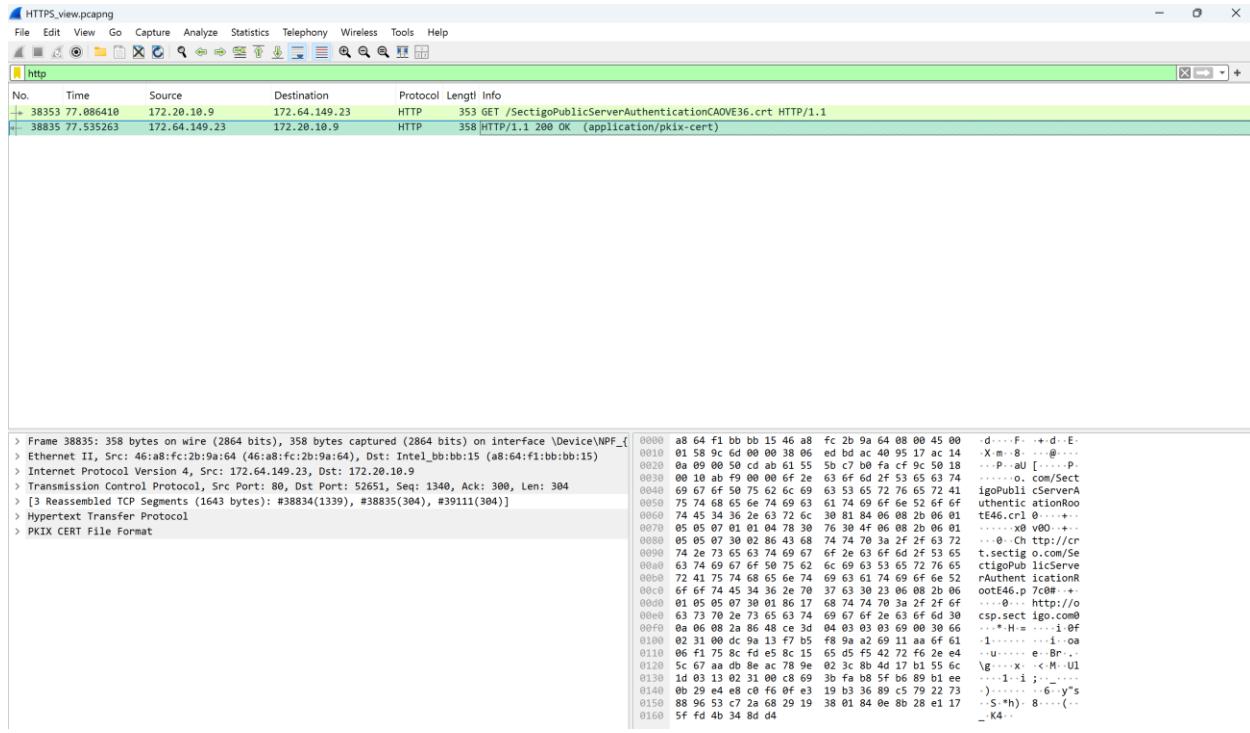


Ilustración 31. Resolución DNS y establecimiento de conexión TCP para la sesión HTTP.

Tras aplicar el filtro **http** en Wireshark, se identificaron los siguientes aspectos del tráfico web generado en la conexión cliente-servidor:

#### Información observada en la capa de aplicación (HTTP)

En la columna **Info** se observan las peticiones y respuestas relacionadas con el servicio web:

- **GET / HTTP/1.1** donde el cliente solicita la página principal del sitio web.
- **HTTP/1.1 200 OK (text/html)** donde el servidor responde con éxito, entregando el recurso solicitado en formato HTML.

#### Protocolo de la capa de transporte

Las peticiones y respuestas HTTP se encapsulan sobre el protocolo **TCP (Transmission Control Protocol)** que se puede evidenciar en la Ilustración anterior.

Esto garantiza una transmisión confiable de los datos, utilizando mecanismos de confirmación (ACK), control de flujo y retransmisiones cuando es necesario.

#### Puertos utilizados

- **Puerto destino (servidor): 80**, reservado para el servicio HTTP.
- **Puerto origen (cliente): 52651**.

Prueba	Método/Respuesta HTTP	Cliente (IP)	Servidor Web (IP)	Protocolo transporte	Puerto destino	Puerto origen
HTTP view.pcap	GET /	172.20.10.7	172.20.10.5	TCP	80	52651

## 5. Análisis del protocolo HTTPS realizando navegación en el sitio de YouTube

Esta prueba puede realizarse desde cualquier equipo que posea conexión a internet, sea un equipo personal o un equipo del laboratorio

**5.1 Utilizando el navegador conéctese al sitio web <https://www.youtube.com/>. Es altamente recomendado tener solamente una única pestaña activa en el navegador con la finalidad de poder realizar la captura de tráfico con la menor interferencia posible.**

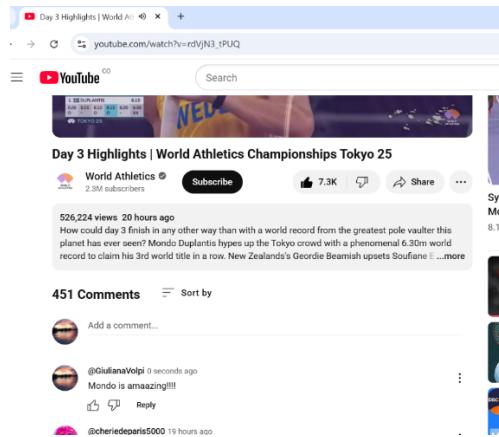
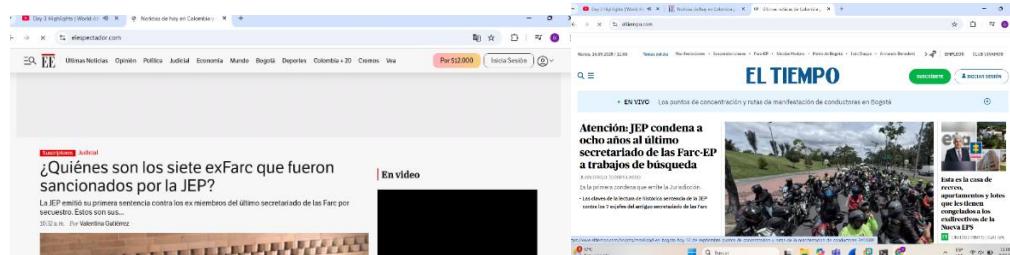


Ilustración 32. Evidencia conexión al sitio web.

Esta imagen evidencia el acceso a youtube junto con la publicación de un comentario, teniendo la sesión iniciada. Esto se evidencia en las siguientes imágenes.

**5.2 Abra una nueva pestaña del navegador, y visite los sitios web: <https://www.elespectador.com>, [https://www.eltiempo.com/](https://www.eltiempo.com), <https://www.uniandes.edu.co/>, y <https://www.bancolombia.com/>. Guarde en un archivo la captura del tráfico generado con el nombre **HTTPS\_view.pcap**.**



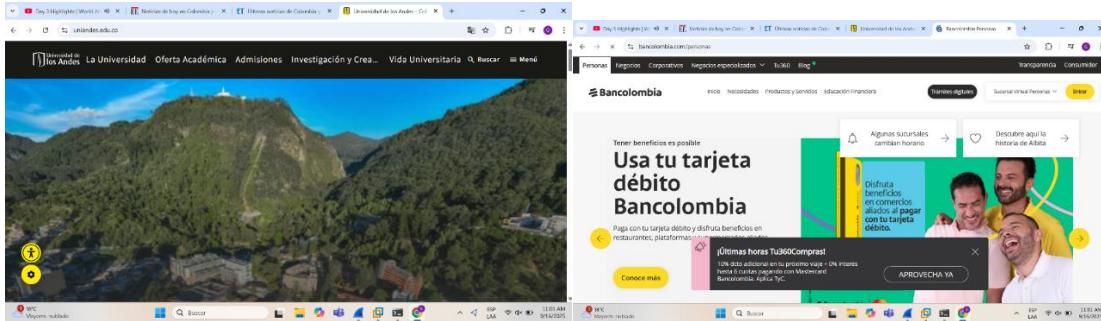


Ilustración 33. Evidencia sitios web requeridos e iniciar sesión .

Este grupo de imágenes son evidencia del ingreso satisfactorio a los sitios web pedidos y teniendo la sesión iniciada en Google.

**5.3 Abra cada uno de los archivos .pcap y realice primero un filtro en la interfaz de Wireshark por el puerto 443 mediante la siguiente instrucción tcp.port==443 y haga click en el botón Apply; a continuación, identifique la siguiente información y consignela en el documento de reporte:**

209 4.043668 172.20.10.9 40.126.62.132 TCP 54 [TCP Dup ACK 288#1] 60340 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0	210 4.051587 172.20.10.9 40.126.62.132 TLSv1.3 529 Client Hello (SNI=login.microsoftonline.com)	211 4.051620 172.20.10.9 40.126.62.132 TCP 529 [TCP Retransmission] 60340 → 443 [PSH, ACK] Seq=1 Ack=1 Win=262144 Len=475
<pre>&gt; Frame 210: 529 bytes on wire (4232 bits), 529 bytes captured (4232 bits) on interface \Device\NPF_{88DE &gt; Ethernet II, Src: Intel_bb:bb:15 (a8:64:f1:bb:bb:15), Dst: 46:a8:fc:2b:9a:64 (46:a8:fc:2b:9a:64) &gt; Internet Protocol Version 4, Src: 172.20.10.9, Dst: 40.126.62.132 &gt; Transmission Control Protocol, Src Port: 60340, Dst Port: 443, Seq: 1, Ack: 1, Len: 475     Source Port: 60340     Destination Port: 443     [Stream index: 12]     [Stream Packet Number: 6]     &gt; [Conversation completeness: Complete, WITH_DATA (47)]     [TCP Segment Len: 475]     Sequence Number: 1      (relative sequence number)     Sequence Number (raw): 11677113593</pre>		

Ilustración 34. Evidencia HTTPS en login de microsoft

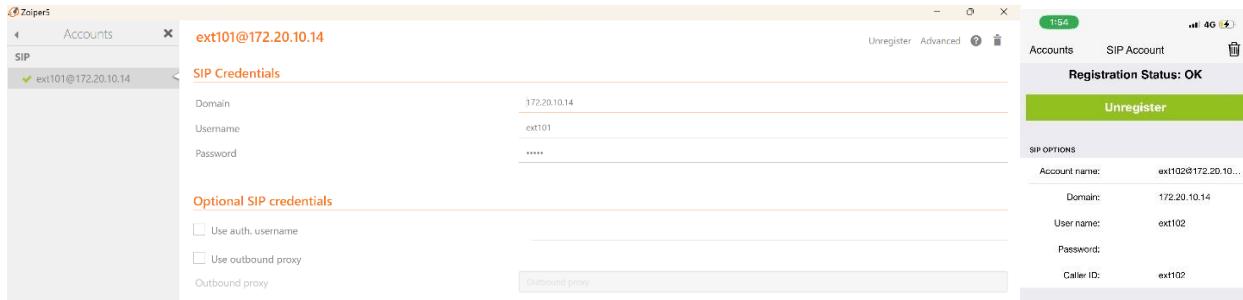
En esta ilustración se observa una conexión HTTPS segura (TLSv1.3) desde un puerto efímero hacia el puerto 443, utilizando TCP para establecer comunicación con el servidor login.microsoftonline.com

- Identifique la información de la capa de aplicación que aparecen en los paquetes capturados que estén relacionados con la navegación segura: Aparece el mensaje Client Hello del protocolo TLSv1.3 con el SNI (Server Name Indication) login.microsoftonline.com, lo que indica navegación segura (HTTPS).
- Identifique el protocolo de la capa de transporte generado por las peticiones al servidor web
- Seguro: El protocolo utilizado es TCP
- Identifique los puertos utilizados por el servicio web seguro: 443 (destino, estándar de HTTPS) y un puerto efímero de origen (60340 en este caso).

## 6. Análisis del protocolo VoIP

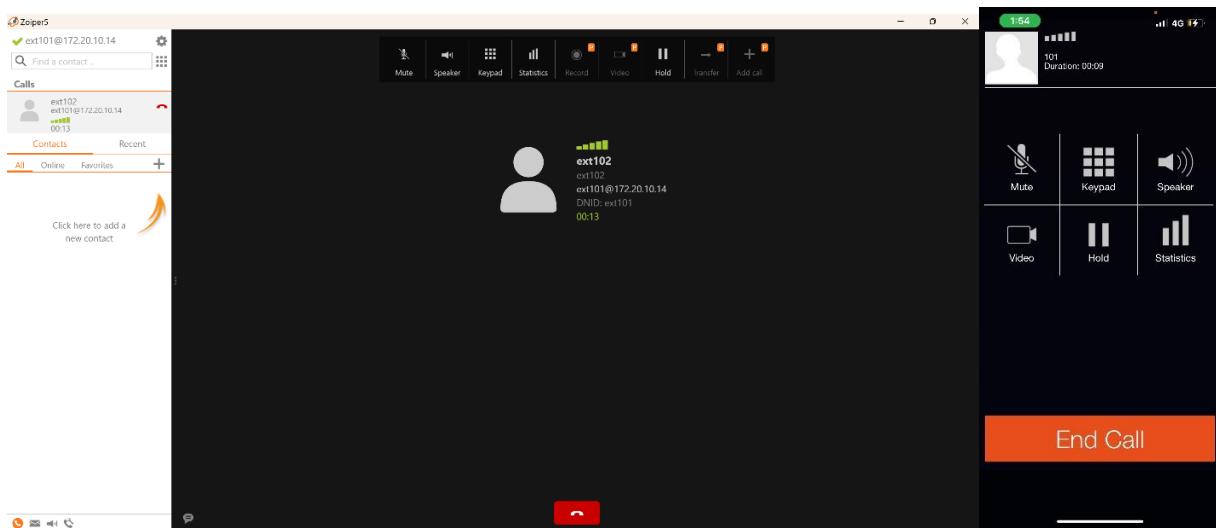
### 6.1 Inicie una llamada entre dos clientes.

En las siguientes dos imágenes de la Ilustración 35, se observa que se registraron dos clientes: uno desde un celular y otro en el computador (un cliente llamado ext102 y el otro ext101 respectivamente).



*Ilustración 35. Configuración de dos clientes en Zoiper: ext101 y ext102*

Luego, en la siguiente ilustración 36 se puede observar la llamada en transcurso desde la interfaz de ambos clientes.



*Ilustración 36. Llamada en transcurso realizada entre los dos clientes*

## **6.2 Abra el archivo .pcap y realice un filtro en la interfaz de Wireshark para distinguir los paquetes provenientes del protocolo**

- Identifique la información de la capa de aplicación que aparece en los paquetes capturados**

No.	Time	Source	Destination	Protocol	Length	Info
409	61.481929	172.20.10.12	172.20.10.14	SIP/SDP	1054	Status: 200 OK (INVITE)
410	61.481949	172.20.10.12	172.20.10.14	SIP/SDP	1054	Status: 200 OK (INVITE)
411	61.483359	172.20.10.14	172.20.10.12	SIP	511	Request: ACK sip:ext101@172.20.10.12:65384
412	61.483373	172.20.10.14	172.20.10.12	SIP	511	Request: ACK sip:ext101@172.20.10.12:65384
413	61.485722	172.20.10.14	172.20.10.1	SIP/SDP	961	Status: 200 OK (INVITE)
414	61.485780	172.20.10.14	172.20.10.1	SIP/SDP	961	Status: 200 OK (INVITE)
415	61.488695	172.20.10.14	172.20.10.12	SIP/SDP	950	Request: INVITE sip:ext101@172.20.10.12:65384, in-dialog
416	61.488715	172.20.10.14	172.20.10.12	SIP/SDP	950	Request: INVITE sip:ext101@172.20.10.12:65384, in-dialog
417	61.529767	172.20.10.1	172.20.10.14	RTP	55	PT=Unassigned, SSRC=0x5AD4F49F, Seq=9876, Time=2729288977
418	61.584822	172.20.10.14	172.20.10.1	SIP/SDP	961	Status: 200 OK (INVITE)
419	61.584835	172.20.10.14	172.20.10.1	SIP/SDP	961	Status: 200 OK (INVITE)
420	61.588503	172.20.10.14	172.20.10.12	SIP/SDP	950	Request: INVITE sip:ext101@172.20.10.12:65384, in-dialog
421	61.588541	172.20.10.14	172.20.10.12	SIP/SDP	950	Request: INVITE sip:ext101@172.20.10.12:65384, in-dialog
422	61.594186	172.20.10.1	172.20.10.14	SIP	452	Request: ACK sip:101@172.20.10.14:5060
423	61.594260	172.20.10.1	172.20.10.14	SIP	452	Request: ACK sip:101@172.20.10.14:5060
424	61.594747	172.20.10.14	172.20.10.1	SIP/SDP	953	Request: INVITE sip:ext102@172.20.10.1:53271;transport=UDP, in-dialog
425	61.594759	172.20.10.14	172.20.10.1	SIP/SDP	953	Request: INVITE sip:ext102@172.20.10.1:53271;transport=UDP, in-dialog
426	61.612208	172.20.10.12	172.20.10.14	SIP	369	Status: 100 Trying
427	61.612217	172.20.10.12	172.20.10.14	SIP	369	Status: 100 Trying
428	61.612489	172.20.10.12	172.20.10.14	SIP/SDP	1054	Status: 200 OK (INVITE)
429	61.612498	172.20.10.12	172.20.10.14	SIP/SDP	1054	Status: 200 OK (INVITE)
430	61.613250	172.20.10.14	172.20.10.12	SIP	511	Request: ACK sip:ext101@172.20.10.12:65384
431	61.613257	172.20.10.14	172.20.10.12	SIP	511	Request: ACK sip:ext101@172.20.10.12:65384
432	61.624087	172.20.10.1	172.20.10.14	SIP/SDP	981	Status: 200 OK (INVITE)
433	61.625651	172.20.10.14	172.20.10.1	SIP	464	Request: ACK sip:ext102@172.20.10.1:53271;transport=UDP
434	61.625660	172.20.10.14	172.20.10.1	SIP	464	Request: ACK sip:ext102@172.20.10.1:53271;transport=UDP
438	61.847982	172.20.10.12	172.20.10.1	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0xA14CDF8E, Seq=44945, Time=2050772499, Mark
439	61.848003	172.20.10.12	172.20.10.1	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0xA14CDF8E, Seq=44945, Time=2050772499, Mark
440	61.854641	172.20.10.1	172.20.10.12	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x5AD4F49F, Seq=9877, Time=2729288977, Mark
441	61.882231	172.20.10.1	172.20.10.12	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x5AD4F49F, Seq=9878, Time=2729289137
442	61.887903	172.20.10.12	172.20.10.1	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0xA14CDF8E, Seq=44946, Time=2050772659

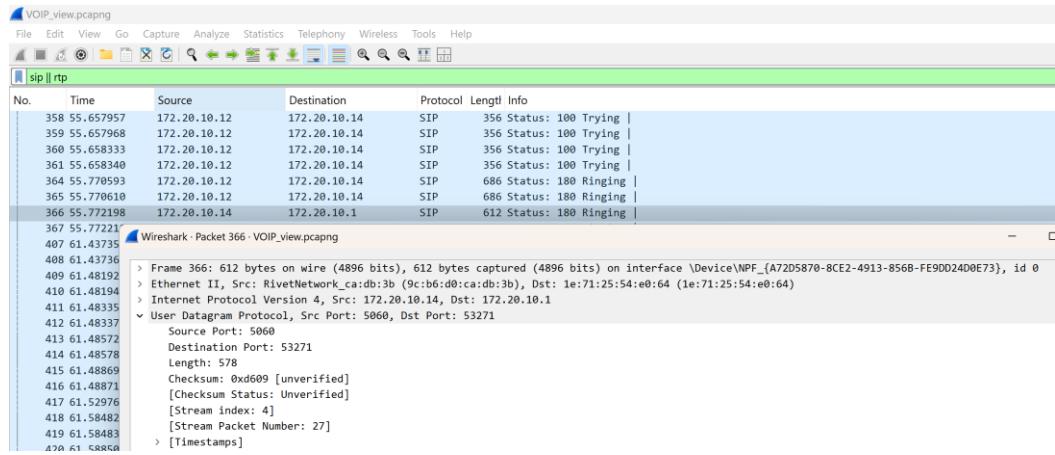
Ilustración 37. Información Capa de Aplicación VOIP

En la captura de tráfico de red de Wireshark de la ilustración 37 se observa la información correspondiente a la capa de aplicación del modelo TCP/IP. Específicamente, se encuentra la información relacionada con una comunicación de voz sobre IP entre los dos usuarios registrados anteriormente, los cuales utilizan Zoiper. Los principales protocolos identificados son SIP (Session Initiation Protocol) y RTP (Real-time Transport Protocol), los cuales cumplen funciones distintas dentro de la comunicación.

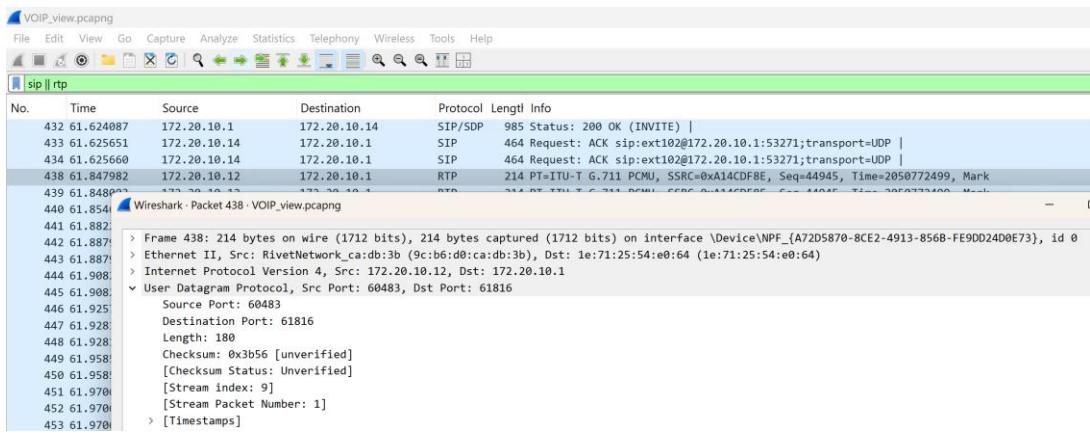
En primer lugar, el protocolo SIP aparece en múltiples mensajes como INVITE, 200 OK y ACK, los cuales conforman el proceso de señalización que permite el establecimiento, mantenimiento y finalización de la llamada realizada entre los dos clientes. En algunos de estos mensajes se incluye información mediante SDP (Session Description Protocol), donde se negocian parámetros como la IP de los participantes, el puerto de comunicación, entre otros. Esto garantiza que ambos extremos de la comunicación puedan intercambiar voz en un formato común y compatible.

Posteriormente, una vez establecida la llamada, se identifican paquetes del protocolo RTP, que transportan la señal de voz digitalizada en tiempo real. En este caso, los paquetes también muestran que se utiliza el códec ITU-T G.711 PCMU. Además, cada paquete incluye información de control como el SSRC (Synchronization Source Identifier), el número de secuencia y la marca de tiempo (timestamp). Todos estos aspectos permiten la sincronización y el orden correcto de los fragmentos de audio para una comunicación fluida.

- **Identifique el protocolo de la capa de transporte utilizado para realizar la llamada y Identifique los puertos utilizados**



*Ilustración 38. Información Capa de Transporte SIP*



*Ilustración 39. Información Capa de Transporte RTP*

Como se observa tanto en la ilustración 38 como la 39, el protocolo de la capa de transporte utilizado por la llamada por VOIP es UDP (User Datagram Protocol). Este protocolo no tiene orientación a conexión y no garantiza la entrega de los paquetes, ni el orden ni la corrección de los paquetes transmitidos. Sin embargo, su bajo tiempo de retardo y simplicidad lo convierten en la opción más adecuada para aplicaciones en tiempo real como VOIP, donde la fluidez y la sincronización de la voz son prioridades por encima de la retransmisión de paquetes perdidos. En particular, los mensajes de SIP viajan encapsulados sobre UDP en el puerto estándar 5060, mientras que el protocolo RTP (Real-time Transport Protocol), encargado de transportar la voz codificada, también hace uso de UDP en puertos dinámicos. Esto se observa en la captura, donde se muestran comunicaciones entre el puerto 5060 y un puerto asignado (53271 en este caso).

## 7. Análisis del protocolo RTMP

### 7.1 Cierre todas las aplicaciones que consuman recursos de red (Navegadores Web, Clientes de Correo Electrónico, Clientes de Mensajería Instantánea, entre otros).

### 7.2 Inicie una transmisión a través del servidor.

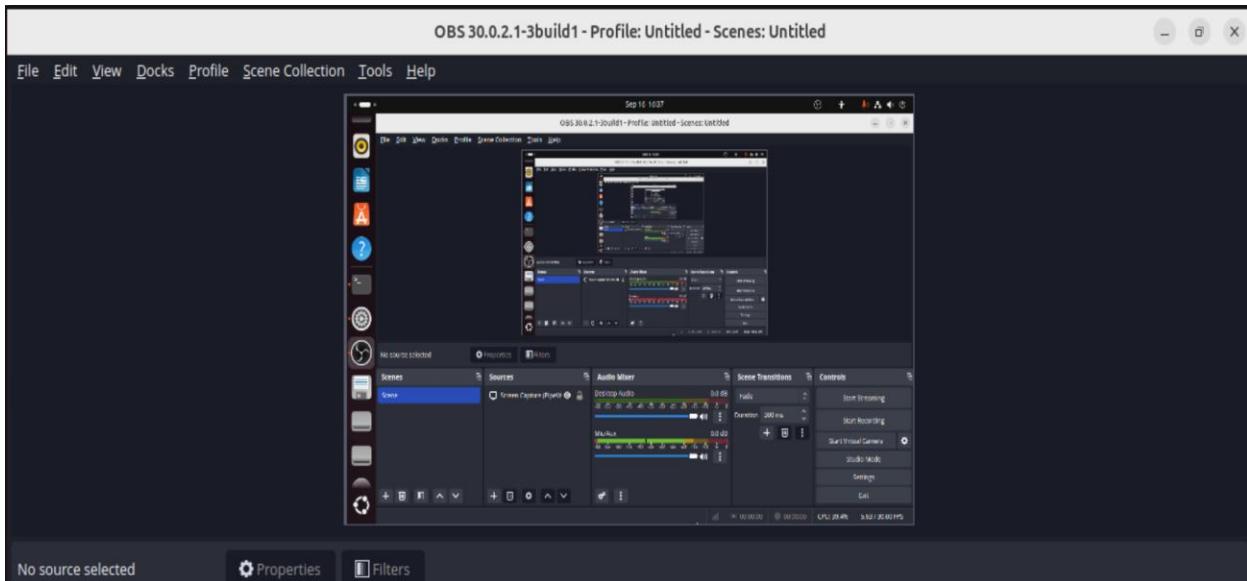


Ilustración 40. Transmisión RTMP

En la Ilustración anterior podemos ver como se inicia la transmisión desde la máquina virtual.

### 7.3 Guarde en un archivo la captura del tráfico generado con el nombre RTMP\_view.pcap.

El archivo de captura quedo guardado en el documento RTMP\_view.pcap.

### 7.4 Abra el archivo .pcap y realice un filtro en la interfaz de Wireshark para distinguir los paquetes Provenientes del protocolo

- Identifique la información de la capa de aplicación que aparece en los paquetes capturados
- Identifique el protocolo de la capa de transporte utilizado para realizar la llamada
- Identifique los puertos utilizados

No.	rtmp	Source	Destination	Protocol	Length	Info
2366	27.416957	172.20.10.5	172.20.10.6	RTMP	155	Handshake C0+C1
2379	27.453394	172.20.10.6	172.20.10.5	RTMP	154	Handshake S0+S1+S2
2384	27.459407	172.20.10.5	172.20.10.6	RTMP	154	Handshake C2
2387	27.464327	172.20.10.5	172.20.10.6	RTMP	277	connect('live')
2388	27.467186	172.20.10.6	172.20.10.5	RTMP	82	Window Acknowledgement Size 5000000
2391	27.472272	172.20.10.6	172.20.10.5	RTMP	301	Set Peer Bandwidth 5000000,Dynamic[Set Chunk Size 4096]_result('NetConnection.Connect.Success')
2397	27.525077	172.20.10.5	172.20.10.6	RTMP	183	Window Acknowledgement Size 5000000[createStream()]
2400	27.528143	172.20.10.6	172.20.10.5	RTMP	107	_result()
2406	27.579413	172.20.10.5	172.20.10.6	RTMP	162	getStreamLength() play('123') Set Buffer Length 1,3000ms
2409	27.581936	172.20.10.6	172.20.10.5	RTMP	84	Stream Begin 1
2412	27.625462	172.20.10.6	172.20.10.5	RTMP	1140	onStatus('NetStream.Play.Start')  RtmpSampleAccess() onMetaData() Audio Data Audio Data
2415	27.718916	172.20.10.6	172.20.10.5	RTMP	457	Audio Data
2418	27.833805	172.20.10.6	172.20.10.5	RTMP	471	Audio Data
2421	27.947027	172.20.10.6	172.20.10.5	RTMP	479	Audio Data
2424	27.976794	172.20.10.6	172.20.10.5	RTMP	489	Audio Data
2427	28.147174	172.20.10.6	172.20.10.5	RTMP	582	Audio Data
2429	28.258325	172.20.10.6	172.20.10.5	RTMP	1514	Audio Data Audio Data Audio Data
2432	28.261311	172.20.10.6	172.20.10.5	RTMP	335	Audio Data
2436	28.295455	172.20.10.6	172.20.10.5	RTMP	519	Audio Data
2439	28.348282	172.20.10.6	172.20.10.5	RTMP	524	Audio Data
2447	28.471104	177.70.10.6	177.70.10.5	RTMP	457	audio.h264

```

> Frame 2366: 155 bytes on wire (1240 bits), 155 bytes captured (1240 bits) on interface \Device\NPF_{0
> Ethernet II, Src: (62:31:d1:47:22:38) (62:31:d1:47:22:38), Dst: Intel_d7:e0:24 (28:6b:35:d7:e0:24)
> Internet Protocol Version 4, Src: 172.20.10.5, Dst: 172.20.10.6
> Transmission Control Protocol, Src Port: 43514, Dst Port: 1935, Seq: 1449, Ack: 1, Len: 89
> Real Time Messaging Protocol (Handshake C0+C1)

0000 28 6b 35 d7 e0 24 62 31 41 47 22 38 08 00 45 00 (k5-5b1-G78-E-
0010 00 84 35 ca 40 00 40 06 98 6d ac 14 0a 05 ac 14 5 @ @ -m-
0020 00 86 a9 fa 07 8f 18 32 a9 67 ee 8d 9c cb 88 18 ..... 2 g-
0030 00 88 a1 f2 00 00 b1 01 08 0a 81 0e c8 19 df 07 ..... b
0040 2c f7 05 af 52 83 0b d2 11 a5 48 88 94 cc 00 05 ,...R....H...
0050 22 55 04 87 11 cd 19 85 10 a9 e9 43 bf 8b 83 91 "U.....C...
0060 98 45 5f bd a1 07 38 05 03 03 99 00 88 d9 b6 36 E_A_0_.....6
0070 56 d4 3c fe c8 ed 83 74 8f 4b 0f 0f c5 12 02 16 V<-t-K...
0080 79 4b 22 a0 54 e5 bc 58 ab d8 c4 10 96 07 08 84 yK~T-X.....
0090 39 34 53 ce 50 96 94 af be ab e0 945-P.....

```

Ilustración 41. RTMP interfaz Wireshark

Tras aplicar el filtro rtmp en Wireshark, se identificaron las siguientes características del tráfico generado durante la transmisión:

### Información de la capa de aplicación

En la columna **Info** se observan diferentes fases del protocolo RTMP:

- **Handshake (C0+C1, S0+S1+S2, C2)**: proceso inicial de establecimiento de la sesión.
- **createStream() y onStatus()**: mensajes de control para iniciar y gestionar la transmisión.
- **Audio Data**: correspondientes a la transmisión del canal de audio.
- **Video Data**: correspondientes al flujo de video.

Esto confirma que el protocolo RTMP se está utilizando para la transmisión en tiempo real de contenido multimedia (audio y video).

### Protocolo de la capa de transporte

RTMP encapsulado como RTMPT (RTMP Tunneled) se transporta sobre el protocolo TCP, como se puede evidenciar en la imagen. Esto garantiza la entrega ordenada y confiable de los segmentos de video y audio durante la transmisión.

### Puertos utilizados

- **Puerto destino (servidor)**: 1935, estándar de RTMP.
- **Puerto origen (cliente)**: 43514.

Prueba	Método/Respuesta HTTP	Cliente (IP)	Servidor RTMP (IP)	Protocolo transporte	Puerto destino	Puerto origen
RTMP	Handshake, Audio Data, Video Data	172.20.10.5	172.20.10.6	TCP	1935	43514

## 8. Bibliografía

- Sección - 2.2 Principles of Network Applications. Computer Networking, a top-down approach. James Kurose, Keith Ross. Addison-Wesley, 6th edición.
- Sección - 2.3 Principles of Network Applications. Computer Networking, a top-down approach. James Kurose, Keith Ross. Addison-Wesley, 6th edición.
- Capítulo 4 Network Analysis Using Wireshark. Yoram Orzach. Packt Publishing