**Laboratorio 1 – Protocolos de la capa de Transporte y Aplicación**



**JUAN CAMILO SARABINO ALEGRÍA**

**Informe número 1 en el curso REDES**

**Profesor:**

**Edwin F. Castillo Q.**

**Universidad del Cauca**

**Facultad de Ingeniería Electrónica y Telecomunicaciones**

**Departamento de Telemática**

**Redes**

**Popayán, febrero 2024**

**JUAN CAMILO SARABINO ALEGRÍA**

**Laboratorio 1 – Protocolos de la capa de Transporte y Aplicación**

Informe No. 1 presentado en el curso de REDES

Estudiante del:

Programa de Ingeniería de Sistemas

Profesor:

Edwin F. Castillo Q

Popayán

2024

Contenido

[Lista de Figuras ii](#_Toc159795866)

[Lista de Tablas iv](#_Toc159795867)

[Laboratorio 1 – Protocolos de la capa de Transporte y Aplicación 1](#_Toc159795868)

[1.- Actividad 1: Configuración de la computadora host del módulo para capturar protocolos de la capa de aplicación 1](#_Toc159795869)

[1.1.- Paso 1. Descargar e instalar Wireshark 1](#_Toc159795870)

[1.2.- Paso 2. Iniciar Wireshark y configurar la interfaz de captura 2](#_Toc159795871)

[2.- Actividad 2: Captura y análisis de la comunicación HTTP entre la computadora host del módulo y un servidor Web. 2](#_Toc159795872)

[2.1.- Paso 1. Iniciar las capturas de Wireshark 2](#_Toc159795873)

[2.2.- Paso 2. Verificar direcciones IP del cliente y el servidor 2](#_Toc159795874)

[2.3.- Paso 3. Iniciar el explorador Web del host del módulo 4](#_Toc159795875)

[2.4.- Paso 4. Detener las capturas de Wireshark y analizar los datos capturados. 5](#_Toc159795876)

[2.5.- Análisis de resultados y respuestas a preguntas planteadas 5](#_Toc159795877)

[3.- Actividad 3: Captura y análisis de la comunicación FTP entre la computadora host del módulo y un servidor Web 12](#_Toc159795878)

[3.1.- Paso 1. Iniciar las capturas de Wireshark 12](#_Toc159795879)

[3.2.- Paso 2. Iniciar el cliente FTP de la línea de comandos host del módulo 12](#_Toc159795880)

[3.3.- Paso 3. Iniciar el explorador Web del host del módulo 13](#_Toc159795881)

[3.4.- Paso 4. Iniciar el cliente FTP de la línea de comandos host del módulo 14](#_Toc159795882)

[3.5.- Paso 5. Modos de transferencia FTP activo y pasivo 16](#_Toc159795883)

[4.- Actividad 4: Reflexión 16](#_Toc159795884)

[5.- Actividad 5: Captura y análisis de la comunicación utilizando el cliente FTP Filezilla 17](#_Toc159795885)

[6.- Actividad 6: Analizar los paquetes DNS (UDP) capturados 18](#_Toc159795886)

[7.- Experiencias de la práctica 21](#_Toc159795887)

[8.- Enlace del video 21](#_Toc159795888)

[9.- Conclusiones 21](#_Toc159795889)

[10.- Referencias 22](#_Toc159795890)

# Lista de Figuras

[Figura 1.1 Interfaz del programa WireShark 1](#_Toc159795891)

[Figura 1.2 Registro de datos en Wireshark 2](#_Toc159795892)

[Figura 2.1 Dirección IP cliente Ethernet 3](#_Toc159795893)

[Figura 2.2 Dirección IP cliente Wifi 3](#_Toc159795894)

[Figura 2.3 Dirección IP Servidor Univirtual 3](#_Toc159795895)

[Figura 2.4 Dirección IP servidor Biblioteca 3](#_Toc159795896)

[Figura 2.5 Dirección IP servidor Univirtual 4](#_Toc159795897)

[Figura 2.6 Dirección IP servidor Biblioteca 4](#_Toc159795898)

[Figura 2.7 Dirección IP servidor Web Local 5](#_Toc159795899)

[Figura 2.8 Solicitud del cliente (SYN) 7](#_Toc159795900)

[Figura 2.9 Respuesta del servidor (SYN, ACK) 7](#_Toc159795901)

[Figura 2.10 Confirmación del cliente (ACK) 7](#_Toc159795902)

[Figura 2.11 Solicitud de finalización (FIN) 8](#_Toc159795903)

[Figura 2.12 Recepción de finalización (ACK) 8](#_Toc159795904)

[Figura 2.13 Solicitud HTTP (GET) 8](#_Toc159795905)

[Figura 2.14 Trama de respuesta a solicitud GET 9](#_Toc159795906)

[Figura 2.15 Datos encapsulados dentro del segmento TCP 9](#_Toc159795907)

[Figura 2.16 Respuesta del servidor detallada para la solicitud GET 10](#_Toc159795908)

[Figura 2.17 Respuesta HTTP/1.1 200 OK del servidor para la solicitud GET 11](#_Toc159795909)

[Figura 3.1 Cliente FTP a ftp.unicauca.edu.co 12](#_Toc159795910)

[Figura 3.3 Accediendo a windows/Documentos 13](#_Toc159795911)

[Figura 3.4 Descargando un archivo 13](#_Toc159795912)

[Figura 3.5 Archivo FTP\_Command\_Line\_Client 13](#_Toc159795913)

[Figura 3.6 Acceso a ftp://ftp.unicauca.edu.co/ desde el explorador de archivos 14](#_Toc159795914)

[Figura 3.7 Archivo descargado 14](#_Toc159795915)

[Figura 3.8 Archivo FTP\_Web\_Browser\_Client 14](#_Toc159795916)

[Figura 3.9 Captura FTP de respuesta 220 15](#_Toc159795917)

[Figura 3.10 Protocolo FTP 15](#_Toc159795918)

[Figura 3.11 Puertos de cliente y servidor FTP 15](#_Toc159795919)

[Figura 3.12 FTP por línea de comando y browser 16](#_Toc159795920)

[Figura 5.1 Intento de conexión al servidor FTP de la Universidad del Cauca 18](#_Toc159795921)

[Figura 6.1 Filtro por dns en Wireshark 18](#_Toc159795922)

[Figura 6.2 UDP por dns en Wireshark para www.google.com 19](#_Toc159795923)

[Figura 6.3 Campos del UDP en la solicitud 19](#_Toc159795924)

[Figura 6.4 Campos del UDP en la respuesta 20](#_Toc159795925)

# Lista de Tablas

[Tabla 2.1 Conexión de cliente por Wifi con Univirtual 5](#_Toc159795926)

[Tabla 2.2 Conexión de cliente por Wifi con Biblioteca 6](#_Toc159795927)

[Tabla 2.3 Conexión de cliente por Ethernet con Web Local 6](#_Toc159795928)

[Tabla 2.4 Información real enviada al servidor Web 9](#_Toc159795929)

# Laboratorio 1 – Protocolos de la capa de Transporte y Aplicación

# 1.- Actividad 1: Configuración de la computadora host del módulo para capturar protocolos de la capa de aplicación

Se procede a realizar la configuración de la computadora host del módulo para capturar protocolos de la capa de aplicación.

## 1.1.- Paso 1. Descargar e instalar Wireshark

Una vez descargado e instalado el programa WireShark, se visualiza una interfaz como la siguiente:

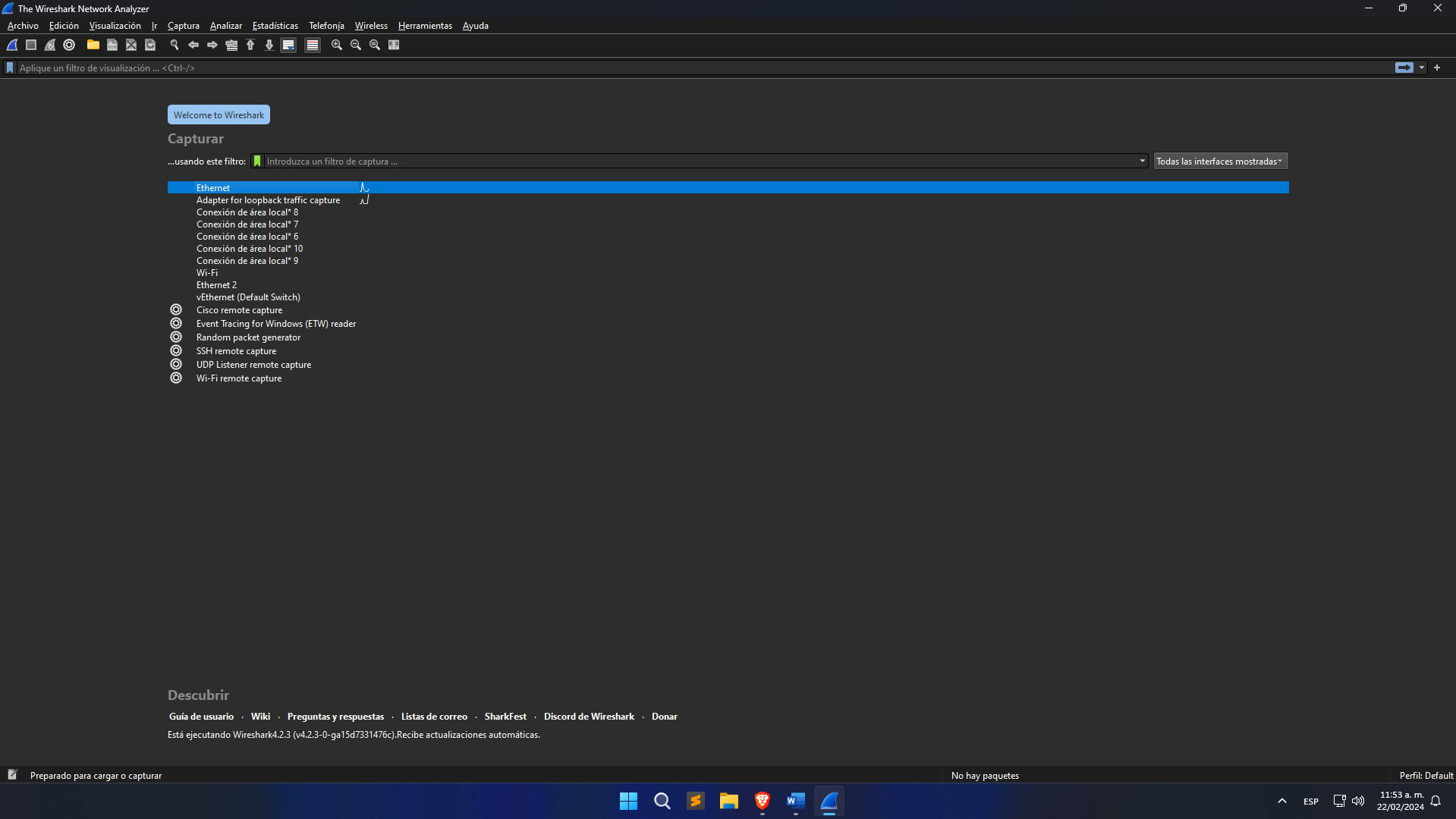


Figura 1.1 Interfaz del programa WireShark

## 1.2.- Paso 2. Iniciar Wireshark y configurar la interfaz de captura

Se comienza a registrar la captura de datos en la interfaz de Ethernet:

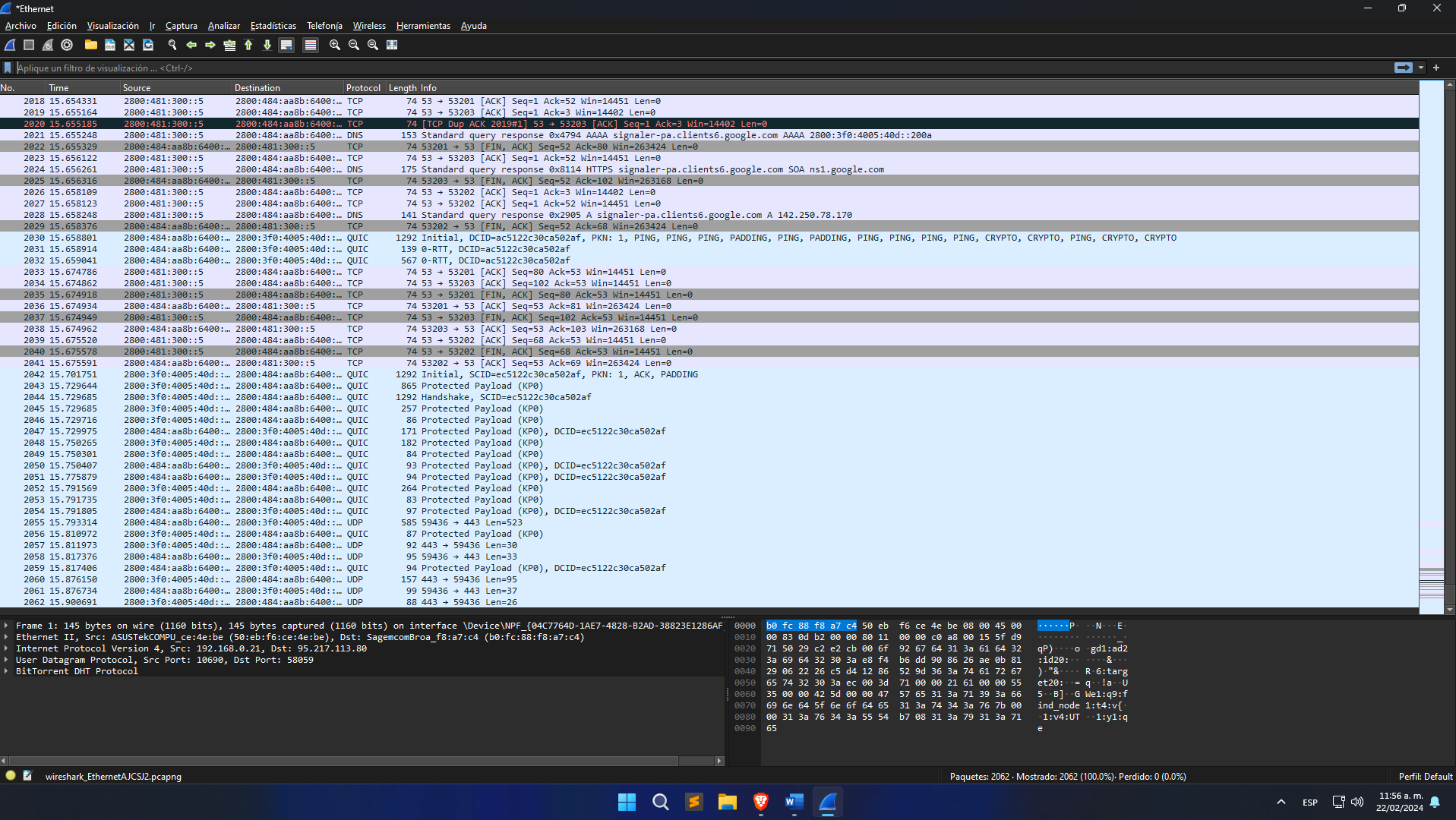


Figura 1.2 Registro de datos en Wireshark

# 2.- Actividad 2: Captura y análisis de la comunicación HTTP entre la computadora host del módulo y un servidor Web.

Se procede a la captura y análisis de la comunicación HTTP entre la computadora host del módulo y un servidor Web.

## 2.1.- Paso 1. Iniciar las capturas de Wireshark

Inicie una captura de Wireshark. Wireshark mostrará capturas basadas en el tipo de paquete. Para ello puede basarse en el video expuesto en la actividad anterior.

## 2.2.- Paso 2. Verificar direcciones IP del cliente y el servidor

* Dirección IP cliente

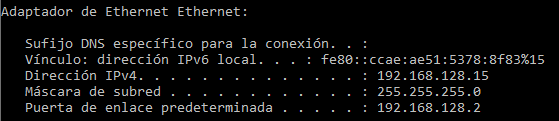


Figura 2.1 Dirección IP cliente Ethernet

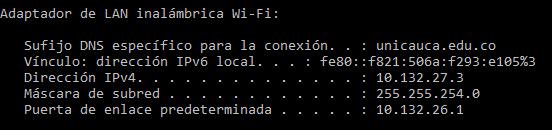


Figura 2.2 Dirección IP cliente Wifi

* Dirección IP del servidor (Univirtual - Universidad del Cauca)

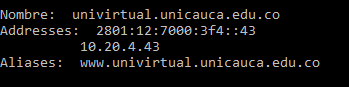


Figura 2.3 Dirección IP Servidor Univirtual

* Dirección IP del servidor (Biblioteca - Universidad del Cauca)



Figura 2.4 Dirección IP servidor Biblioteca

* Dirección IP del servidor (web local proporcionado por docente)

192.168.128.2

## 2.3.- Paso 3. Iniciar el explorador Web del host del módulo

Se eligió el explorador Web “Brave” y se procedió a iniciar en el host

1. Se iniciaron las capturas de Wireshark.
2. En el explorador Brave se accedió a <https://univirtual.unicauca.edu.co/>.
3. Se actualizó la página de Univirtual, efectivamente no hubo cambios en la pantalla del cliente, sin embargo, se obtuvo las siguientes tramas:

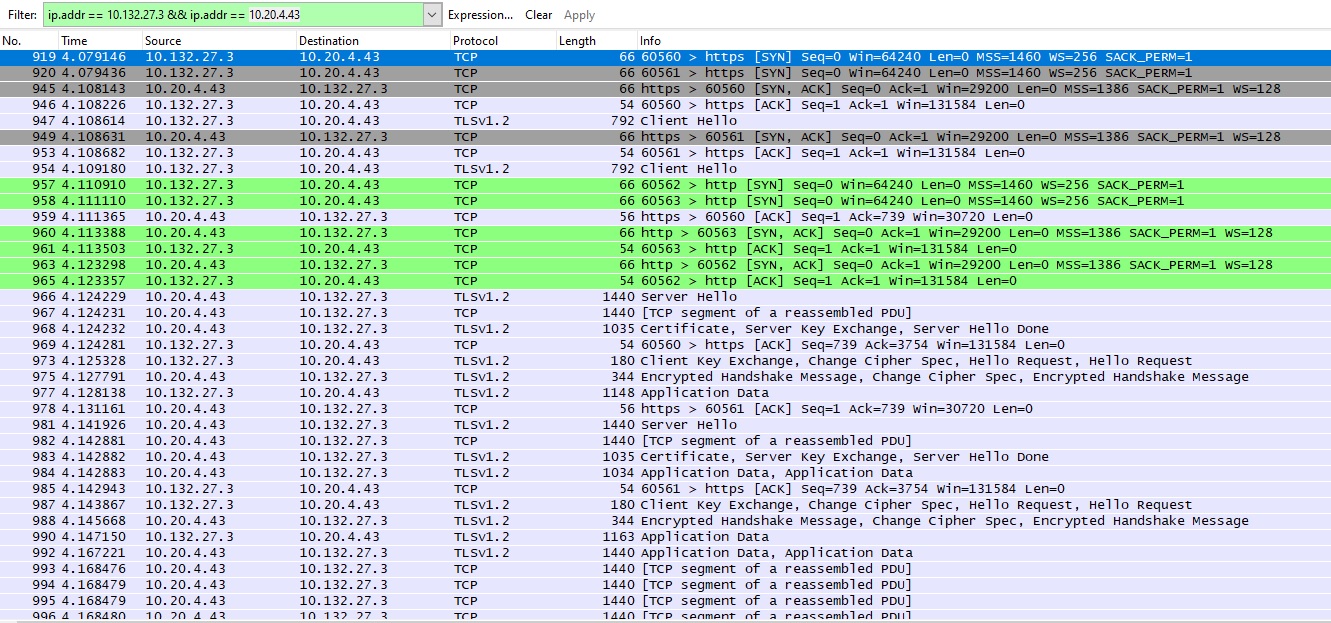


Figura 2.5 Dirección IP servidor Univirtual

Tener más tramas de lo normal para realizar una conexión con Univirtual supone la captura de más tramas que indiquen las solicitudes adicionales al actualizar la página.

1. Se abrió un segundo explorador Web y se procedió a conectarse a <http://biblio.unicauca.edu.co/>. Se obtuvo las siguientes tramas:

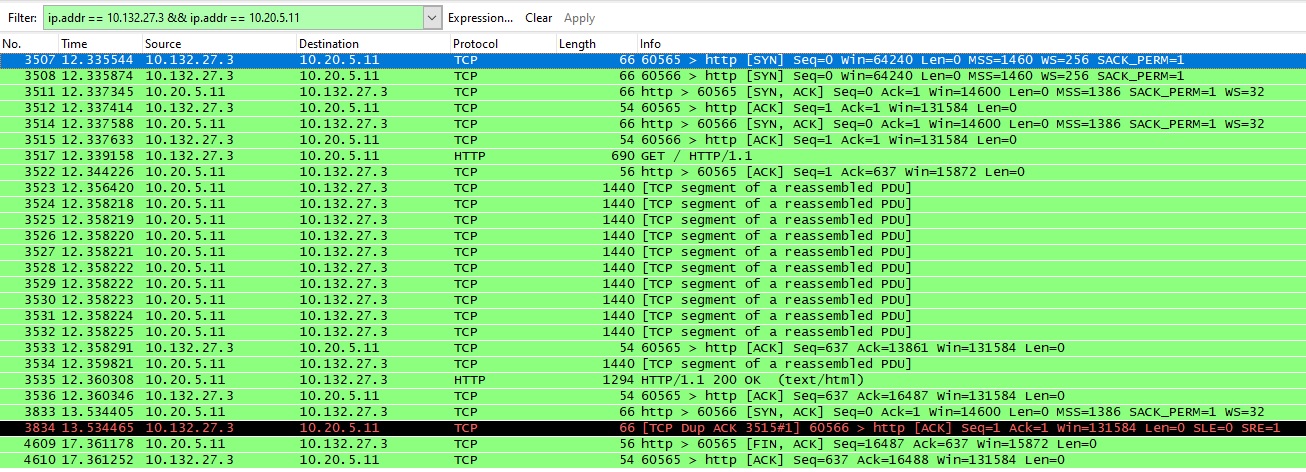


Figura 2.6 Dirección IP servidor Biblioteca

1. En un tercer explorador Web, se accedió a la dirección <http://192.168.128.2/redes/public/login> y se obtuvo las siguientes tramas:

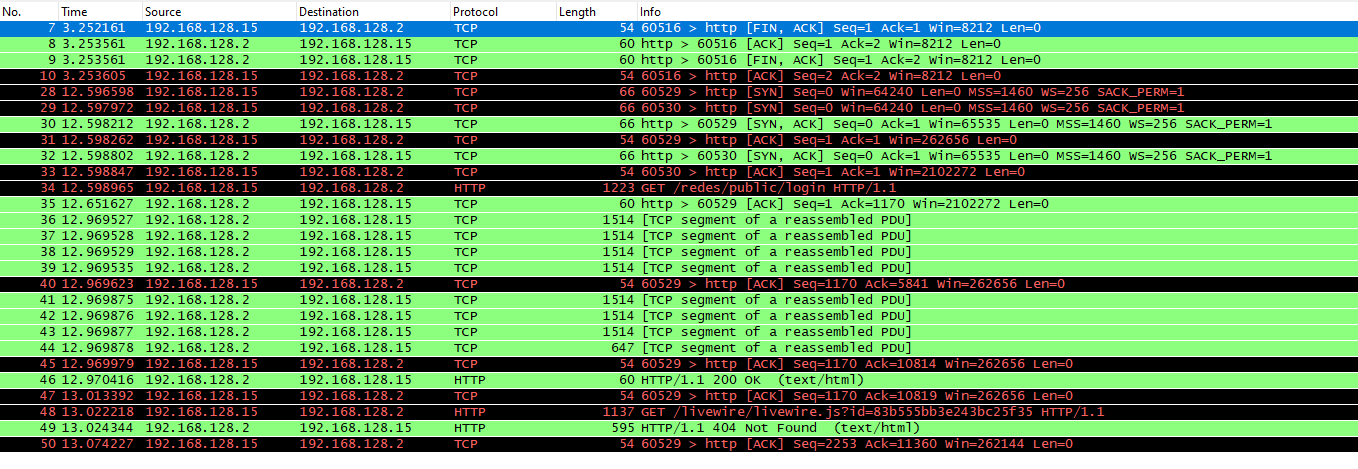


Figura 2.7 Dirección IP servidor Web Local

## 2.4.- Paso 4. Detener las capturas de Wireshark y analizar los datos capturados.

Por último, se detuvieron las capturas de Wireshark y se cerraron los exploradores Web.

**NOTA:** Debido a problemas de conexión en la sala que se realizó el laboratorio, las IP del cliente son diferentes para Ethernet y Wifi. La conexión a Univirtual y a la biblioteca se realizó por conexión Wifi y la conexión al servidor Web Local se realizó por Ethernet.

## 2.5.- Análisis de resultados y respuestas a preguntas planteadas

* 1. Complete la siguiente tabla con la información presentada en la sesión HTTP, hacer los mismo para el acceso a los tres servicios web:

|  |  |
| --- | --- |
| **Dirección IP del explorador Web (cliente)** | 10.132.27.3 |
| **Dirección IP del servidor Web** | 10.20.4.43 |
| **Protocolo de la capa de transporte (UDP/TCP)** | TCP |
| **Número de puerto del explorador Web** | 60560 |
| **Número de puerto del servidor Web** | 443 |

Tabla 2.1 Conexión de cliente por Wifi con Univirtual

|  |  |
| --- | --- |
| **Dirección IP del explorador Web (cliente)** | 10.132.27.3 |
| **Dirección IP del servidor Web** | 10.20.5.11 |
| **Protocolo de la capa de transporte (UDP/TCP)** | TCP |
| **Número de puerto del explorador Web** | 60565 |
| **Número de puerto del servidor Web** | 80 |

Tabla 2.2 Conexión de cliente por Wifi con Biblioteca

|  |  |
| --- | --- |
| **Dirección IP del explorador Web (cliente)** | 192.168.128.15 |
| **Dirección IP del servidor Web** | 192.168.128.2 |
| **Protocolo de la capa de transporte (UDP/TCP)** | TCP |
| **Número de puerto del explorador Web** | 60516 |
| **Número de puerto del servidor Web** | 80 |

Tabla 2.3 Conexión de cliente por Ethernet con Web Local

* 1. ¿Qué computadora inició la sesión HTTP y cómo lo hizo?, responder la pregunta teniendo en cuenta la información teórica de los capítulos 1-4 del curso.

La computadora que inicia la sesión HTTP es la computadora cliente, ya que esta inicia la sesión al enviar una solicitud HTTP al servidor (en el caso de Univirtual envía una solicitud HTTPS), en los tres casos de conexión, la capa de transporte usa el protocolo TCP, el cual permite una conexión confiable. TCP utiliza la conexión de tres vías.

* 1. En TCP se establecen las conexiones usando el protocolo de acuerdo a tres vías (three-way handshake), utilice los resultados de las capturas para evidenciar este proceso. Explique en detalle cada vía en cada caso (soporte la explicación con capturas de pantalla de la vía correspondiente).

Primero el cliente envía un paquete con el bit SYN con un número de secuencia inicial, en este caso con la solicitud a Univirtual se envía un número de secuencia inicial 0, el indicador SYN le dice al servidor que el cliente quiere establecer una conexión:

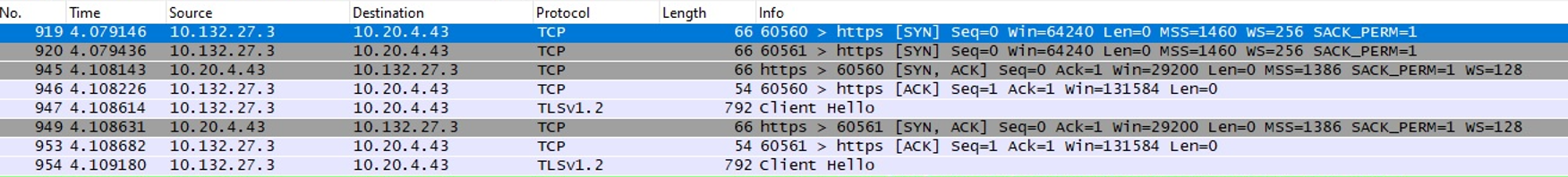


Figura 2.8 Solicitud del cliente (SYN)

Al recibir la solicitud, el servidor envía una respuesta con los bits SYN y ACK, donde el número de secuencia SYN es propio del servidor, en este caso el servidor envía 0, y el bit ACK es el número de secuencia del cliente aumentado en 1:

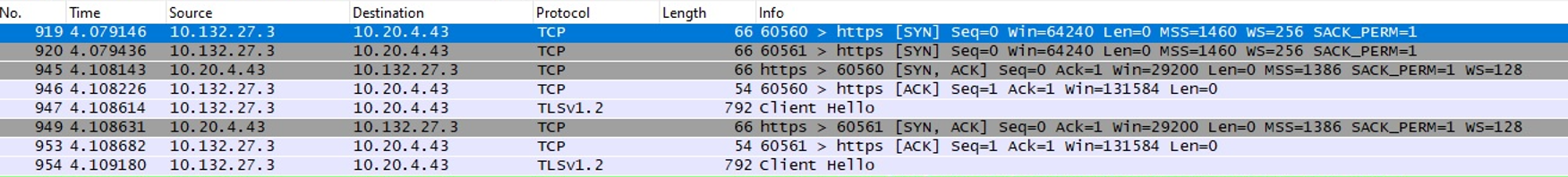


Figura 2.9 Respuesta del servidor (SYN, ACK)

Por último, el cliente responde con el bit ACK activo incrementando en 1 el número de secuencia que el servidor envió anteriormente:

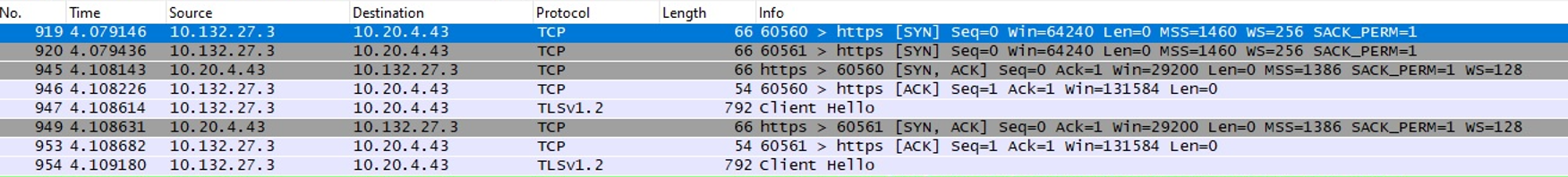


Figura 2.10 Confirmación del cliente (ACK)

Una vez realizados estos pasos, se estableció una sesión y pueden intercambiar información.

* 1. ¿Qué computadora señaló inicialmente un fin a la sesión HTTP y cómo lo hizo?, explicar y dejar evidencia.

Se realiza un cierre de conexión en 4 pasos. Inicialmente el servidor envía un segmento con los bits FIN y ACK, solicitando al cliente el cierre de la sesión ya que ha completado la solicitud del cliente:

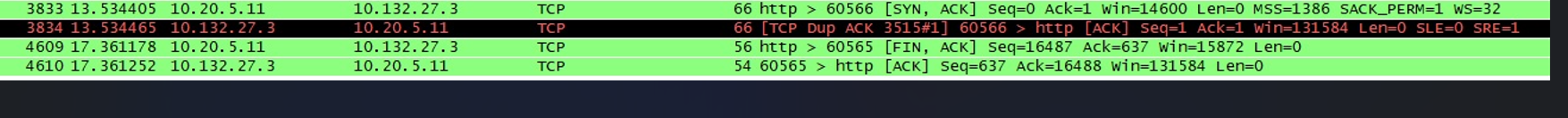


Figura 2.11 Solicitud de finalización (FIN)

El cliente envía al servidor un acuse de recibo, para confirmar que ha confirmado la solicitud de FIN del servidor:

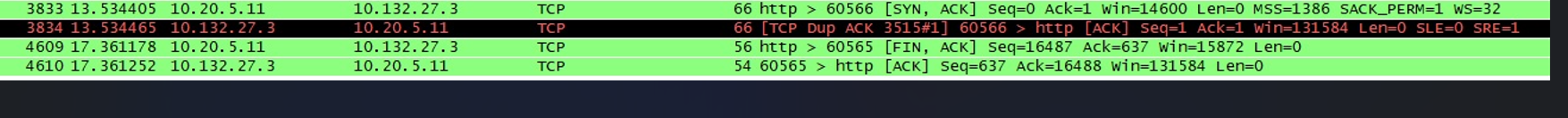


Figura 2.12 Recepción de finalización (ACK)

A pesar que la imagen tomada termina hasta ese paso, lo siguiente que ocurre es el tercer paso para las finalizaciones entre el cliente y el servidor, en el cual el cliente habiendo completado la transmisión de datos, también envía un segmento con el bit FIN. El cuarto paso, el servidor envía un segmento con el bit ACK para confirmar la recepción del segmento FIN del cliente.

* 1. Resalte la primera línea del protocolo HTTP, una solicitud GET (Obtener) del explorador Web. Vaya a la segunda ventana de Wireshark (doble clic a la línea en mención) para examinar los protocolos en capas. Si es necesario, expanda los campos.



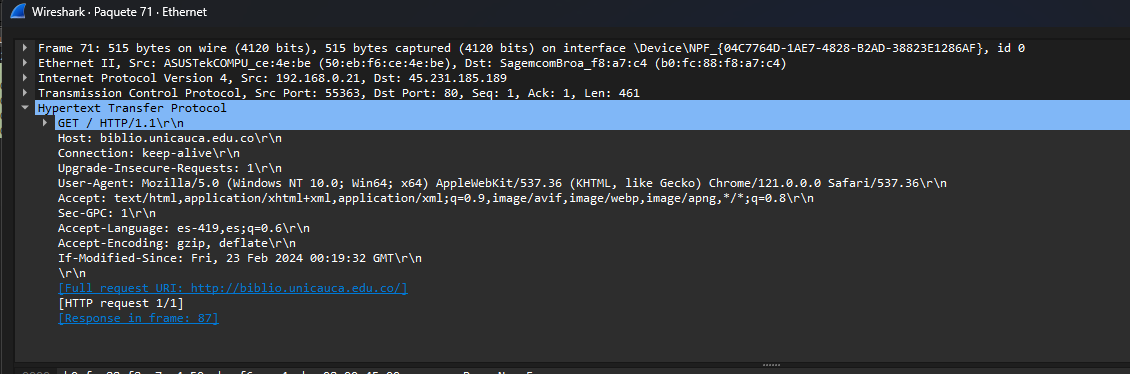


Figura 2.13 Solicitud HTTP (GET)

* 1. ¿Qué protocolo se lleva (encapsulado) dentro del segmento TCP?, dejar evidencia.

Se deja encapsulado el protocolo HTTP, a pesar de no tener evidencia del laboratorio realizado en clase, se puede realizar una conexión a la biblioteca virtual, si se realiza doble click sobre la solicitud GET se puede observar que además de los datos del encabezado, se tienen los datos HTTP correspondientes a la respuesta del servidor:



Figura 2.14 Trama de respuesta a solicitud GET

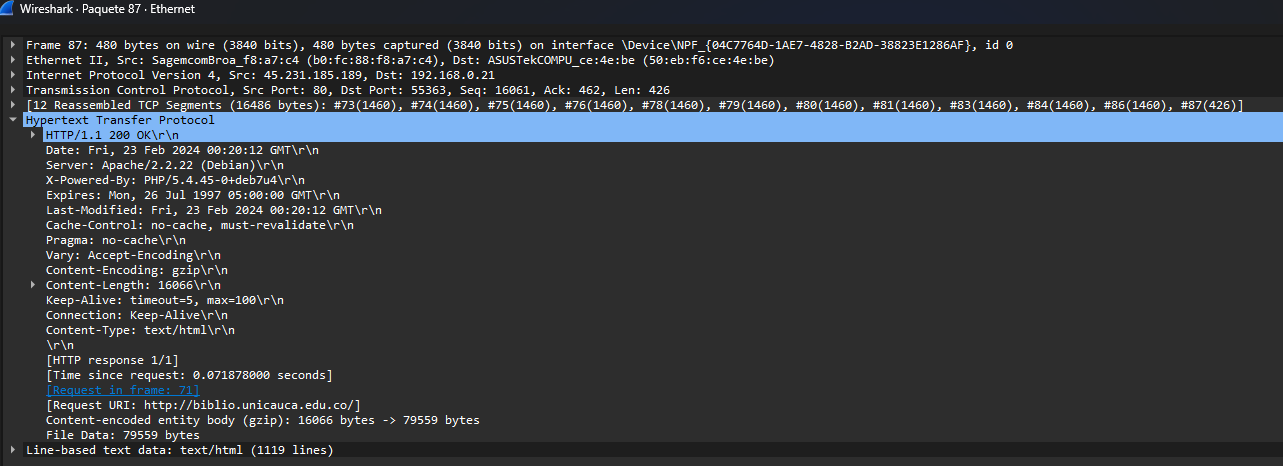


Figura 2.15 Datos encapsulados dentro del segmento TCP

* 1. Expanda el último registro de protocolo y cualquier subcampo. Ésta es la información real enviada al servidor Web. Complete la siguiente tabla utilizando la información del protocolo.

|  |  |
| --- | --- |
| **Versión del protocolo** | HTTP/1.1 |
| **Método de solicitud** | GET |
| **Solicitud URI** | / |
| **Idioma** | es-419, es;q=0.6 |

Tabla 2.4 Información real enviada al servidor Web

Es importante mencionar que la solicitud URI es la ruta para el documento solicitado. En el primer explorador, la ruta es el directorio raíz del servidor Web. Aunque no se solicitó ninguna página, algunos servidores Web están configurados para mostrar un archivo predeterminado, si está disponible.

El servidor Web responde con el próximo paquete HTTP (text/html). Una respuesta para el explorador Web es posible porque el servidor Web (1) comprende el tipo de solicitud y (2) tiene que devolver un archivo.

Nota importante: Los crackers a veces envían solicitudes desconocidas o dañadas a servidores Web para intentar detener el servidor o poder acceder a la línea de comando del servidor. Además, una solicitud para una página Web desconocida da como resultado un mensaje de error.

* 1. Resalte la respuesta del servidor Web y luego vaya a la segunda ventana (la del medio). Abra todos los subcampos de HTTP colapsados. Observe la información que devuelve el servidor. En esta respuesta, sólo hay unas pocas líneas de texto (las respuestas del servidor Web pueden contener miles o millones de bytes). El explorador Web comprende los datos de la ventana del explorador y los formatea correctamente.

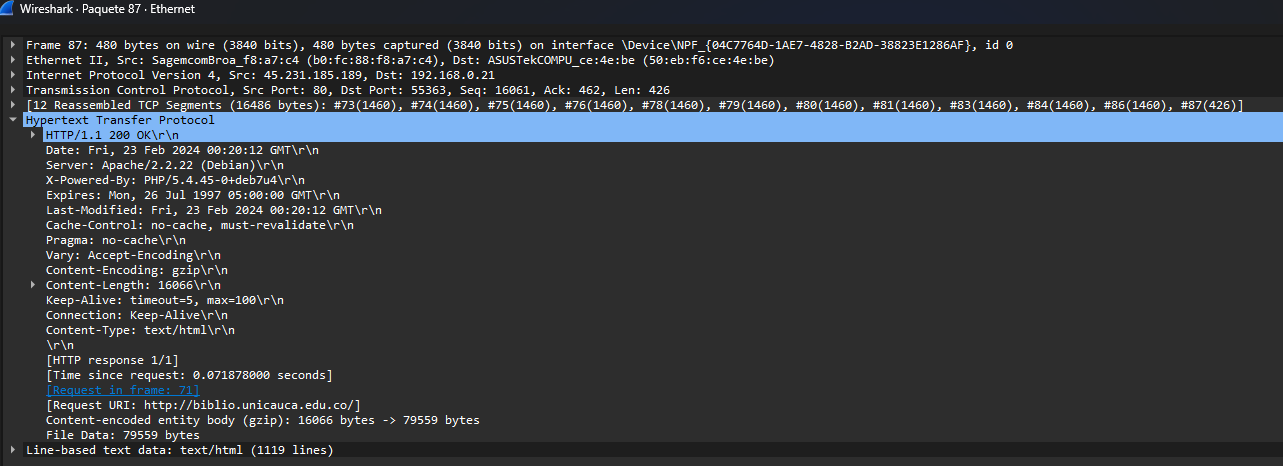


Figura 2.16 Respuesta del servidor detallada para la solicitud GET

* 1. ¿Cuál es la respuesta del servidor Web para la solicitud GET del cliente Web?, explicar y dejar evidencia.

La respuesta del servidor Web es HTTP/1.1 200 OK lo cual indica que fue la solicitud fue exitosa y el servidor devuelve una respuesta.

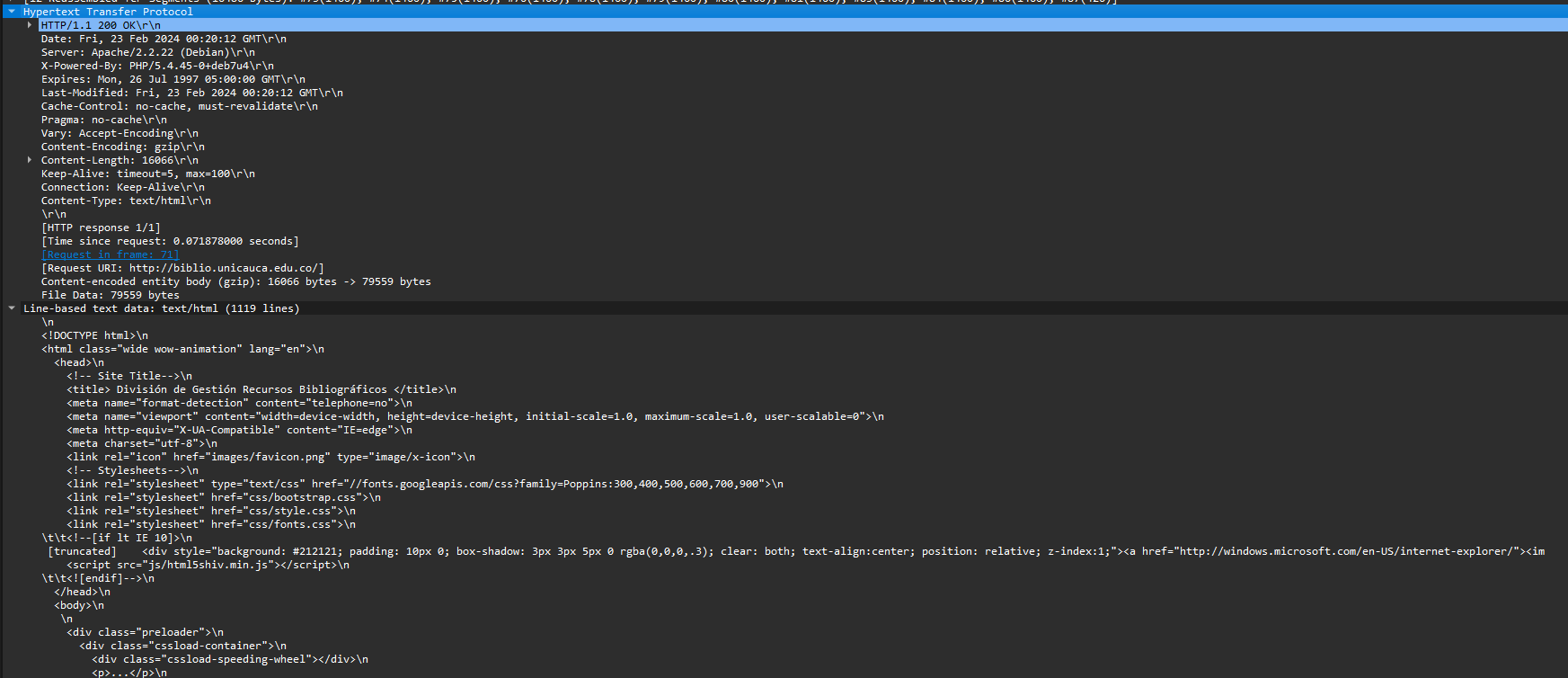


Figura 2.17 Respuesta HTTP/1.1 200 OK del servidor para la solicitud GET

* 1. ¿Qué significa esta respuesta?

El servidor al devolver una respuesta con código de estado 200 (OK) indica que está devolviendo los datos solicitados, posterior a esto incluye encabezados como Date, Server, Content, que proporcionan información adicional del contenido que mandó el servidor además del contenido, en este caso la página, que es un documento HTML.

* 1. Desplácese hacia abajo de la ventana superior de Wireshark hasta que se muestre la segunda sesión de HTTP, actualizada. El significado de la acción de actualización se encuentra en la respuesta del servidor, 304 Not Modified (304 No modificado), localice esta línea en Wireshark.

Con un paquete simple devuelto para la solicitud inicial de GET y para la actualización, el ancho de banda utilizada es mínimo. Sin embargo, para una respuesta inicial que contenga millones de bytes, un simple paquete de respuesta puede generar un significativo ahorro de ancho de banda.

Debido a que esta página Web ha sido guardada en la caché del cliente Web, la solicitud GET contenía las siguientes instrucciones adicionales para el servidor Web.



Localice esta información en Wireshark y dejar la respectiva evidencia.

# 3.- Actividad 3: Captura y análisis de la comunicación FTP entre la computadora host del módulo y un servidor Web

## 3.1.- Paso 1. Iniciar las capturas de Wireshark

1. Se inician las capturas de Wireshark.

## 3.2.- Paso 2. Iniciar el cliente FTP de la línea de comandos host del módulo

1. Se inicia el cliente FTP de la línea de comandos host del módulo.

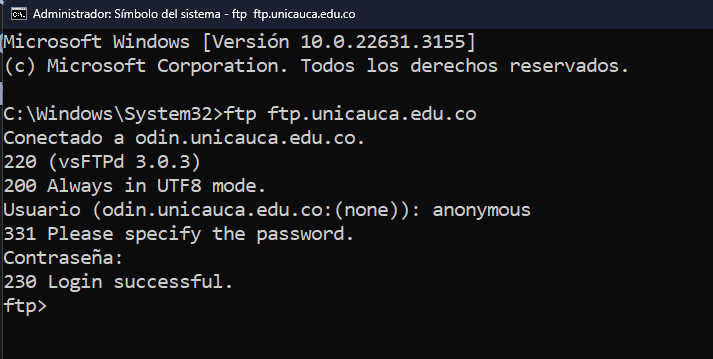
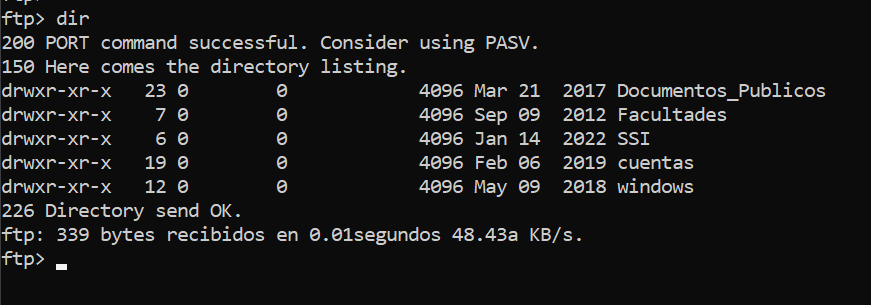


Figura 3.1 Cliente FTP a ftp.unicauca.edu.co

1. No se presentan problemas.
2. Se observan los comandos al introducir la palabra “help”.
3. Se ingresa “dir” para observar el contenido del directorio actual.

Figura 3.2 Contenido del directorio actual

1. Se dirige a la dirección windows/Documentos y se descarga un archivo.

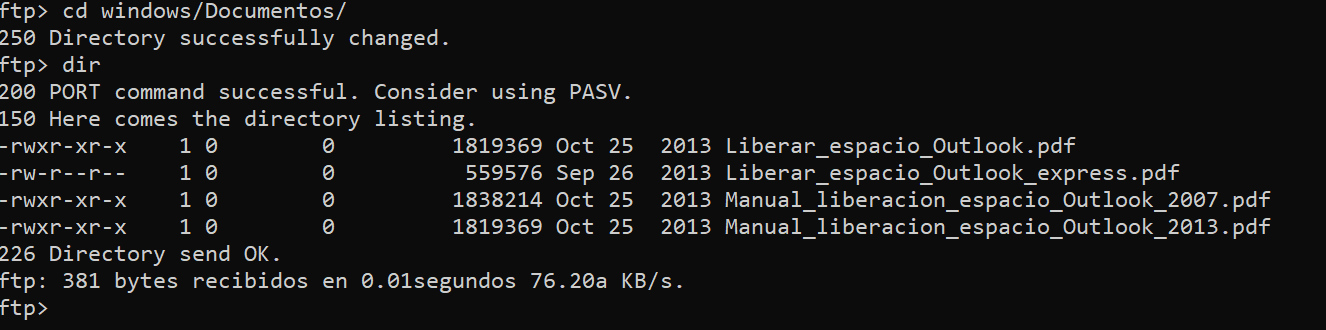


Figura 3.3 Accediendo a windows/Documentos

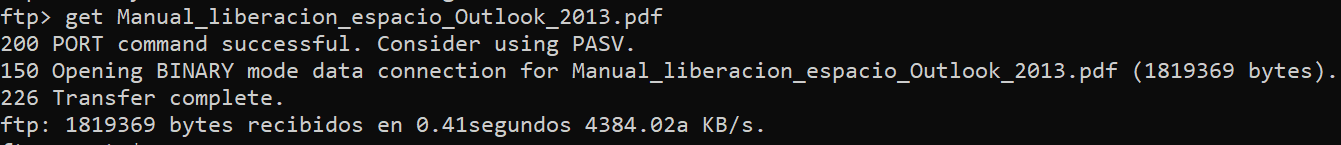


Figura 3.4 Descargando un archivo

1. Se cierra la ventana de línea de comandos.
2. Se detienen las capturas y se guardan como FTP\_Command\_Line\_Client.

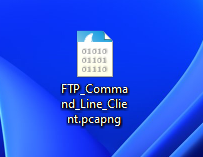


Figura 3.5 Archivo FTP\_Command\_Line\_Client

## 3.3.- Paso 3. Iniciar el explorador Web del host del módulo

1. Se inicia nuevamente la captura Wireshark.
2. Se accede desde el explorador de archivos a la URL <ftp://ftp.unicauca.edu.co/>.

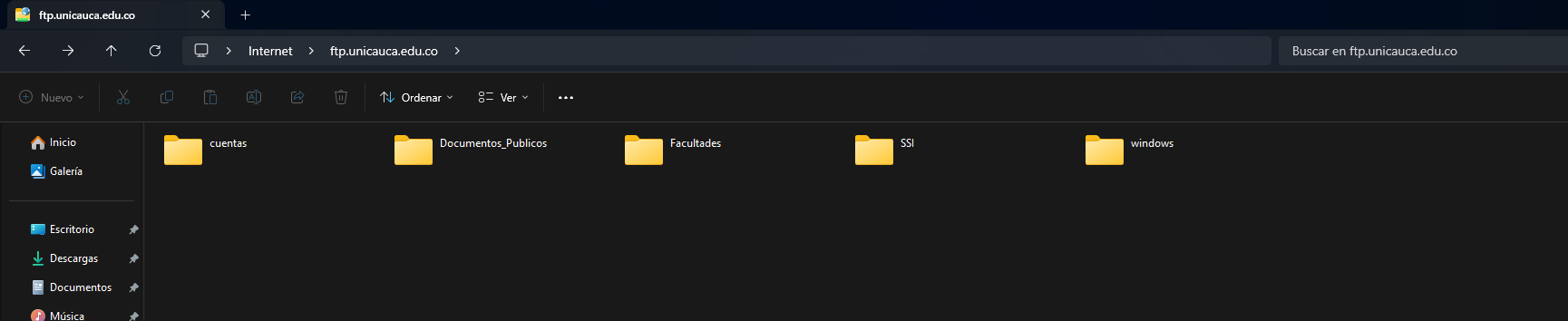


Figura 3.6 Acceso a ftp://ftp.unicauca.edu.co/ desde el explorador de archivos

1. Utilizando el explorador se procede a entrar a windows/Documentos y se guarda el archivo Liberar\_espacio\_Outlook.pdf.

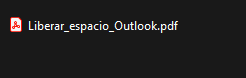


Figura 3.7 Archivo descargado

1. Se cierra el explorador.
2. Se detienen las capturas Wireshark y se guarda como FTP\_Web\_Browser\_Client.



Figura 3.8 Archivo FTP\_Web\_Browser\_Client

## 3.4.- Paso 4. Iniciar el cliente FTP de la línea de comandos host del módulo

1. Se abre la captura FTP\_Web\_Browser\_Client.
2. Se selecciona la captura FTP que es la primera transmisión del protocolo FTP. Respuesta: 220.

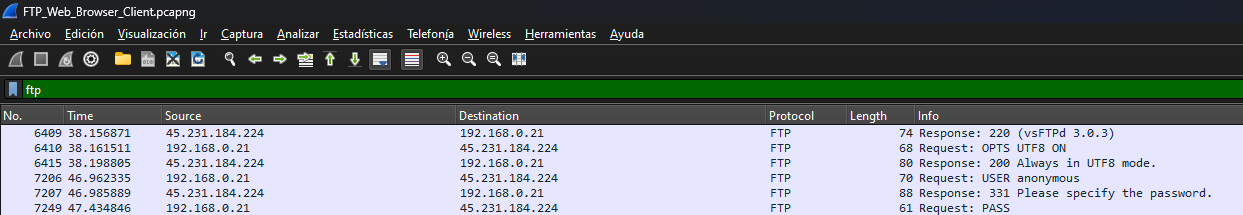


Figura 3.9 Captura FTP de respuesta 220

1. Se expande el protocolo FTP.

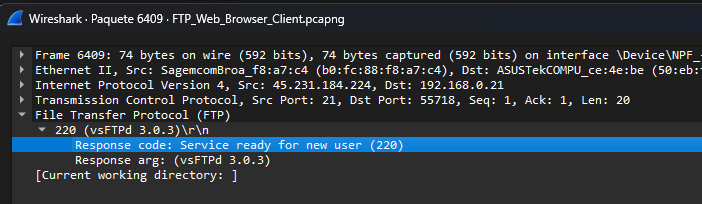


Figura 3.10 Protocolo FTP

1. ¿Cuál es la respuesta 220 del servidor FTP?

Indica que el servicio está listo para un nuevo usuario.

1. Cuando el servidor FTP emitió una Respuesta: 331. Especifique la contraseña. ¿Cuál fue la respuesta del explorador Web?

El explorador web solicita la contraseña, la contraseña era vacía.

1. ¿Qué número de puerto utiliza el cliente FTP para conectarse al puerto 21 del servidor FTP?

Utiliza el puerto 55718

1. ¿Cuál es el número de puerto de Datos FTP utilizado por el servidor FTP?

Utiliza el número de puerto 21



Figura 3.11 Puertos de cliente y servidor FTP

1. Se abre la captura FTP\_Command\_Line\_Client y se comparan los clientes por línea de comando y browser.

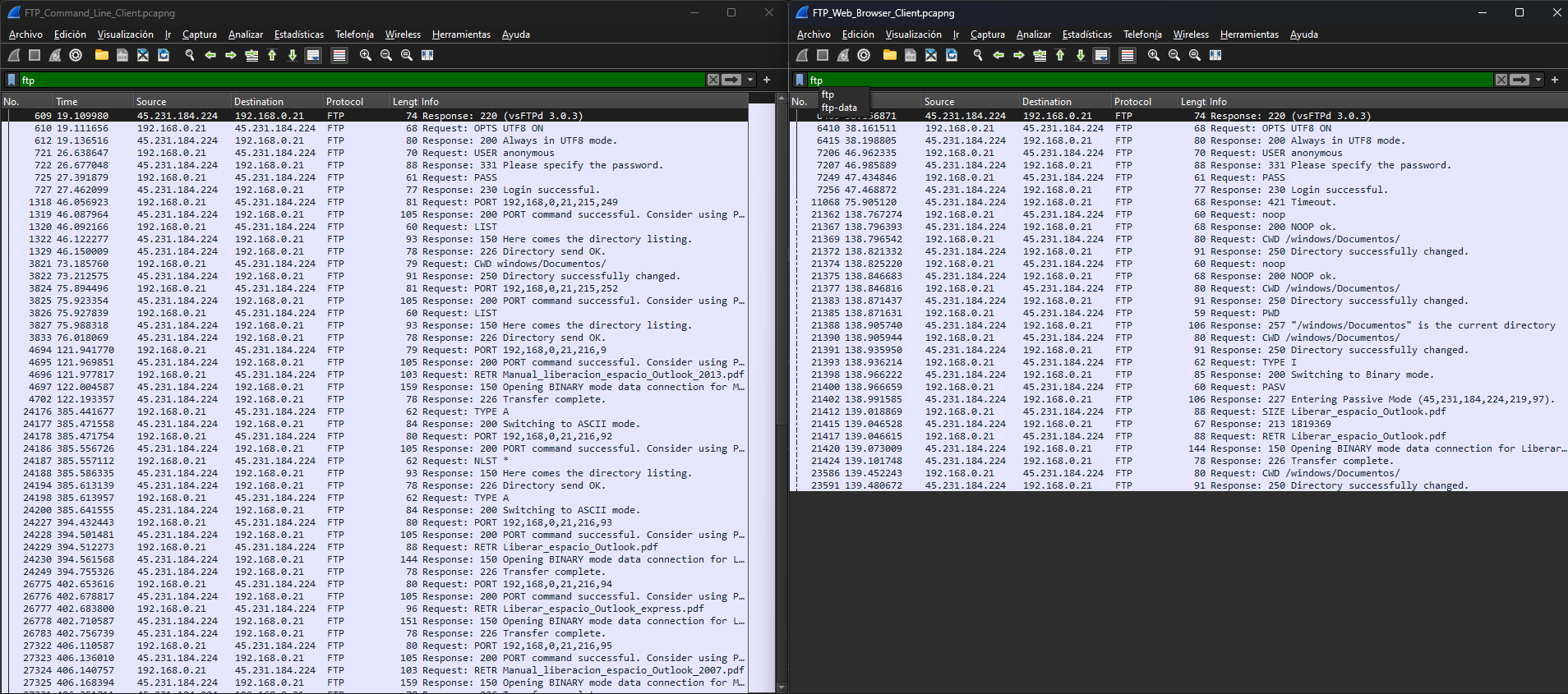


Figura 3.12 FTP por línea de comando y browser

## 3.5.- Paso 5. Modos de transferencia FTP activo y pasivo

La principal diferencia entre estos dos modos radica en cómo se establece la conexión de datos entre el cliente y el servidor FTP.

* En el modo activo: El cliente FTP inicia la conexión de datos con el servidor FTP utilizando un puerto de origen aleatorio no privilegiado (mayor a 1024).

El servidor FTP responde a la conexión del cliente utilizando el puerto de control FTP (puerto 21) como puerto de destino.

* En el modo pasivo: El servidor FTP inicia la conexión de datos con el cliente FTP utilizando un puerto de destino aleatorio no privilegiado (mayor a 1024). El cliente FTP responde a la conexión del servidor utilizando el puerto de control FTP (puerto 21) como puerto de origen.

# 4.- Actividad 4: Reflexión

Tanto HTTP (Protocolo de Transferencia de Hipertexto) como FTP (Protocolo de Transferencia de Archivos) utilizan diferentes métodos para finalizar una sesión de comunicación entre el cliente y el servidor.

* Finalización en HTTP: En HTTP/1.0, por defecto, la conexión se cierra después de que el servidor haya enviado la respuesta.

En HTTP/1.1, la conexión se mantiene viva por defecto (a menos que se especifique lo contrario con la cabecera Connection: close). Sin embargo, el cliente o el servidor pueden decidir cerrar la conexión después de un cierto tiempo de inactividad o después de enviar un determinado número de solicitudes.

La sesión HTTP finaliza una vez que la conexión se cierra o después de que ambas partes (cliente y servidor) decidan cerrarla explícitamente.

* Finalización en FTP: La comunicación también se realiza mediante la emisión de comandos desde el cliente y las respuestas del servidor. Una vez que se ha completado la transferencia de archivos o cualquier otra operación solicitada, la sesión puede considerarse como finalizada.

Comando QUIT: Para finalizar explícitamente la sesión FTP, el cliente emite el comando QUIT al servidor. El servidor responde con un código de respuesta indicando que la sesión se ha cerrado correctamente.

Cierre de Conexión: Después de que se envía el comando QUIT y se recibe la respuesta correspondiente, la conexión FTP se cierra.

La conexión puede cerrarse por el cliente o el servidor una vez que se completa el intercambio de comandos y datos.

# 5.- Actividad 5: Captura y análisis de la comunicación utilizando el cliente FTP Filezilla

No fue posible realizar esta parte del laboratorio debido a que al seguir las instrucciones no se pudo establecer una comunicación con el servidor FTP de la Universidad del Cauca. El principal problema que se encontró es que al tratar de poner las credenciales en el software FileZilla, la conexión se intentaba establecer, pero sin éxito.

Probablemente esto es debido a que se está accediendo de manera incorrecta, a pesar de seguir los pasos indicado en el video indicado dentro del material de apoyo, al tratar de obtener las credenciales en la página CDmon, no fue posible obtener una respuesta a pesar de haber creado una cuenta en la página para este propósito, por lo tanto, solo se deja evidencia del intento de conexión en FileZilla:

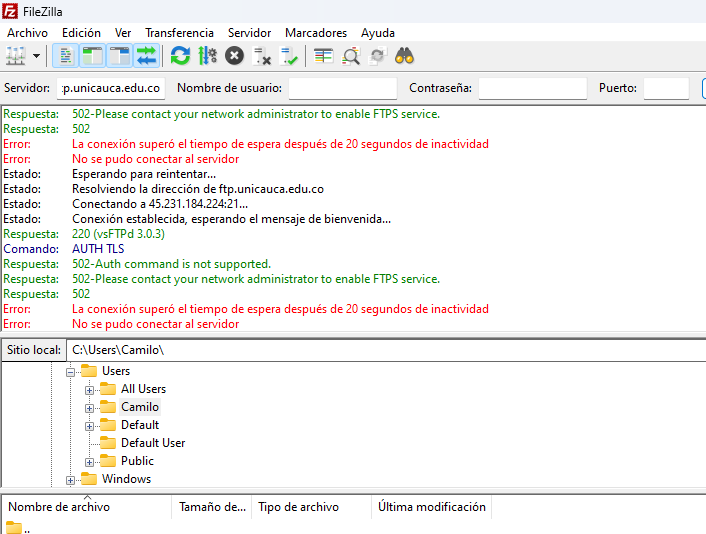


Figura 5.1 Intento de conexión al servidor FTP de la Universidad del Cauca

# 6.- Actividad 6: Analizar los paquetes DNS (UDP) capturados

1. Se abilita la captura en Wireshark y se accede a [www.google.com](http://www.google.com) en el explorador Web.
2. Se detiene la captura en Wireshark.
3. Se filtra por dns.

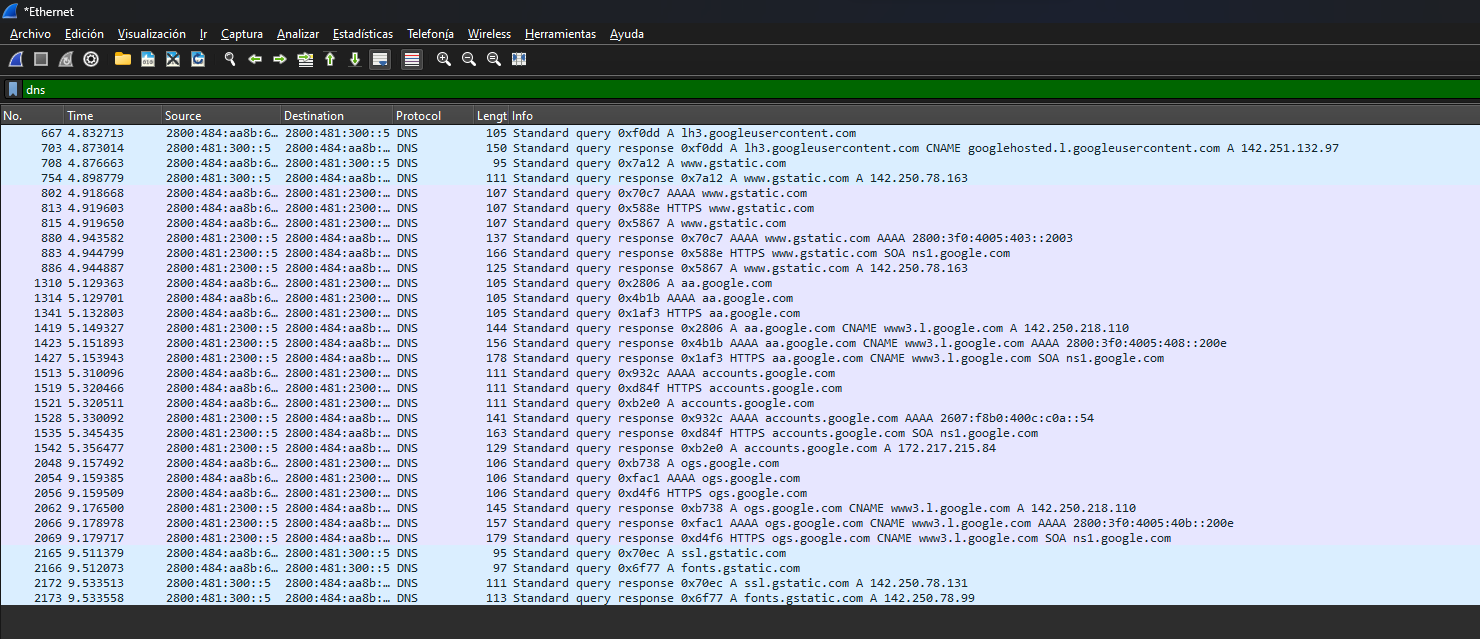


Figura 6.1 Filtro por dns en Wireshark

1. Se ubica el paquete standard query y “A [www.google.com](http://www.google.com)”.
2. Se examina el UDP capturado por Wireshark.

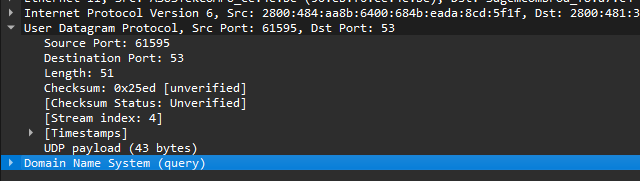


Figura 6.2 UDP por dns en Wireshark para [www.google.com](http://www.google.com)

1. Se amplia el protocolo y se observan los cuatro campos indicados.

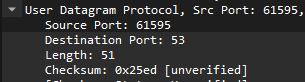


Figura 6.3 Campos del UDP en la solicitud

* Puerto de origen = 61595, es el puerto del remitente del paquete UDP.
* Puerto de destino = 53, es el puerto del receptor, 53 es el número estándar para consultas DNS.
* Longitud = 51, es la longitud total del paquete incluyendo encabezado y consulta.
* Checksum = 0x25ed unverified, es un valor para detectar errores en el paquete UDP durante la transmisión, no verificado significa que Wireshark no ha verificado la suma de comprobación del paquete.

1. La longitud del paquete es 51 bytes, considerando que la cabecera en UDP es de 8 bytes, entonces los 43 bytes restantes serían de la consulta DNS.
2. Observe el datagrama de respuesta (DNS→Cliente) y responda:

* ¿Observe la cantidad de bytes y compárelo con el tamaño del datagrama de la consulta, cual es más grande? Dejar evidencia

La longitud del datagrama de respuesta de 96 bytes es más grande:

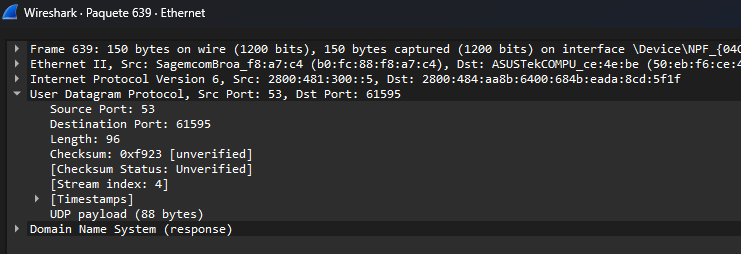


Figura 6.4 Campos del UDP en la respuesta

* ¿Cuál fue el comportamiento de los números de puerto? Observe que hay solo cuatro campos. Identifique y explique cada uno de dichos campos. Dejar evidencia del resultado obtenido.

Puerto de origen: 53, Puerto de destino: 61595

* ¿Cuáles son los beneficios de utilizar UDP en lugar de TCP como protocolo de transporte para servicios o protocolos como DNS o DHCP o por ejemplo streaming de audio y video?

UDP ofrece beneficios como menor sobrecarga y latencia en comparación con TCP. Esto se debe a que UDP no tiene mecanismos de control de flujo, retransmisión de paquetes o control de congestión, lo que lo hace más eficiente para aplicaciones que pueden tolerar cierta pérdida de datos.

* ¿Cuáles son los beneficios de utilizar TCP en lugar de UDP como protocolo de transporte para protocolos de mail (POP3, SMTP), conexiones remotas (Telnet, SSH) y/o transferencia de archivos (FTP)?

TCP ofrece beneficios como la entrega confiable de datos, el control de flujo y la retransmisión de paquetes en caso de pérdida.

1. ¿Cuáles son los beneficios de utilizar UDP en lugar de TCP como protocolo de transporte para servicios o protocolos como DNS o DHCP o por ejemplo streaming de audio y video?

UDP es preferible sobre TCP en situaciones donde la entrega confiable de datos no es crítica y se valora la velocidad, la eficiencia y la simplicidad de implementación. Esto lo hace especialmente adecuado para servicios y protocolos como DNS, DHCP y streaming de audio y video, donde la velocidad y la latencia baja son prioritarias sobre la integridad absoluta de los datos.

1. ¿Cuáles son los beneficios de utilizar TCP en lugar de UDP como protocolo de transporte para protocolos de mail (POP3, SMTP), conexiones remotas (Telnet, SSH) y/o transferencia de archivos (FTP)?

TCP es preferible sobre UDP en situaciones donde la entrega confiable de datos, el control de flujo, el establecimiento de conexión y la seguridad son prioritarios. Esto lo hace adecuado para protocolos de correo electrónico, conexiones remotas y transferencia de archivos, donde la integridad y la seguridad de los datos son fundamentales.

# 7.- Experiencias de la práctica

Se encontraron algunos problemas al momento de realizar la práctica, por ejemplo:

* Falta de conexión por Ethernet al momento de realizar las actividades que requerían de un trabajo en clase para poder capturar las tramas en la red de la Universidad.
* Obligar a estar conectado a la red de la Universidad para realizar algunos puntos, debido a lo anterior las capturas de pantalla de las tramas no daban suficiente información para realizar la práctica completa, ya que la información debe ser consistente en cuanto direcciones IP y otros aspectos.
* Algunos pasos de la práctica son ambiguos, en especial la parte de FileZilla, no se pudo realizar esta parte de la práctica ya que no se contaba con la información adecuada para este proceso.

Aunque estos “problemas” fueron hallados durante la realización de la práctica, en general estuvo interesante y organizada. Algunas soluciones para estos inconvenientes pueden ser tan simples como mantener más contacto con el profesor, al ser una práctica tan larga, sería adecuado dividir la práctica en dos sesiones, donde en la segunda sesión los estudiantes puedan llegar con preguntas acerca de la práctica y poder solucionarlas.

# 8.- Enlace del video

<https://youtu.be/z51YRrJVWpU>

# 9.- Conclusiones

Se han explorado varios temas relacionados con redes y protocolos de Internet. Se ha discutido sobre la transferencia de archivos a través de FTP, examinado los protocolos HTTP y FTP en detalle, y analizado la terminación de sesiones en estos protocolos utilizando herramientas como Wireshark, incluso comparando los modos de transferencia activo y pasivo en FTP, así como los beneficios de TCP y UDP en diferentes contextos de aplicación.

En general, se ha abordado una amplia gama de temas relacionados con la comunicación de datos en redes, desde la transferencia de archivos hasta los protocolos de correo electrónico, conexiones remotas y streaming de medios. Todo esto demuestra la importancia de las conexiones de las redes en las que se basa la comunicación a nivel mundial y la importancia especial de conocer el funcionamiento de los diferentes protocolos de la capa de transporte.

# 10.- Referencias

**1** https://[www.netacad.com/es](http://www.netacad.com/es)

2 https://www.webempresa.com/blog/ftp-y-uso-de-filezilla.html

3 https://raiolanetworks.es/blog/manual-filezilla-cliente-ftp/

4 https://www.youtube.com/watch?v=szdiWw\_CWKE