

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
Lista de Tablas	iv
Laboratorio 1 – Protocolos de la capa de Transporte y Aplicación	1
1.- Actividad 1: Configuración de la computadora host del módulo para capturar protocolos de la capa de aplicación	1
1.1.- Paso 1. Descargar e instalar Wireshark	1
1.2.- Paso 2. Iniciar Wireshark y configurar la interfaz de captura	2
2.- Actividad 2: Captura y análisis de la comunicación HTTP entre la computadora host del módulo y un servidor Web.....	2
2.1.- Paso 1. Iniciar las capturas de Wireshark	2
2.2.- Paso 2. Verificar direcciones IP del cliente y el servidor	2
2.3.- Paso 3. Iniciar el explorador Web del host del módulo	4
2.4.- Paso 4. Detener las capturas de Wireshark y analizar los datos capturados.	5
2.5.- Análisis de resultados y respuestas a preguntas planteadas.....	5
3.- Actividad 3: Captura y análisis de la comunicación FTP entre la computadora host del módulo y un servidor Web.....	12
3.1.- Paso 1. Iniciar las capturas de Wireshark	12
3.2.- Paso 2. Iniciar el cliente FTP de la línea de comandos host del módulo	12
3.3.- Paso 3. Iniciar el explorador Web del host del módulo	13
3.4.- Paso 4. Iniciar el cliente FTP de la línea de comandos host del módulo	14
3.5.- Paso 5. Modos de transferencia FTP activo y pasivo	16
4.- Actividad 4: Reflexión	16
5.- Actividad 5: Captura y análisis de la comunicación utilizando el cliente FTP Filezilla ..	17
6.- Actividad 6: Analizar los paquetes DNS (UDP) capturados	18
7.- Experiencias de la práctica	21
8.- Enlace del video.....	21
9.- Conclusiones.....	21
10.- Referencias	22

Lista de Figuras

Figura 1.1 Interfaz del programa WireShark	1
Figura 1.2 Registro de datos en Wireshark	2
Figura 2.1 Dirección IP cliente Ethernet.....	3
Figura 2.2 Dirección IP cliente Wifi	3
Figura 2.3 Dirección IP Servidor Univirtual.....	3
Figura 2.4 Dirección IP servidor Biblioteca.....	3
Figura 2.5 Dirección IP servidor Univirtual.....	4
Figura 2.6 Dirección IP servidor Biblioteca.....	4
Figura 2.7 Dirección IP servidor Web Local	5
Figura 2.8 Solicitud del cliente (SYN)	7
Figura 2.9 Respuesta del servidor (SYN, ACK)	7
Figura 2.10 Confirmación del cliente (ACK).....	7
Figura 2.11 Solicitud de finalización (FIN)	8
Figura 2.12 Recepción de finalización (ACK)	8
Figura 2.13 Solicitud HTTP (GET)	8
Figura 2.14 Trama de respuesta a solicitud GET.....	9
Figura 2.15 Datos encapsulados dentro del segmento TCP	9
Figura 2.16 Respuesta del servidor detallada para la solicitud GET	10
Figura 2.17 Respuesta HTTP/1.1 200 OK del servidor para la solicitud GET	11
Figura 3.1 Cliente FTP a ftp.unicauca.edu.co	12
Figura 3.3 Accediendo a windows/Documentos	13
Figura 3.4 Descargando un archivo.....	13
Figura 3.5 Archivo FTP_Command_Line_Client.....	13
Figura 3.6 Acceso a ftp://ftp.unicauca.edu.co/ desde el explorador de archivos.....	14
Figura 3.7 Archivo descargado	14
Figura 3.8 Archivo FTP_Web_Browser_Client.....	14
Figura 3.9 Captura FTP de respuesta 220.....	15
Figura 3.10 Protocolo FTP	15
Figura 3.11 Puertos de cliente y servidor FTP	15
Figura 3.12 FTP por línea de comando y browser	16
Figura 5.1 Intento de conexión al servidor FTP de la Universidad del Cauca.....	18

Figura 6.1 Filtro por dns en Wireshark.....	18
Figura 6.2 UDP por dns en Wireshark para www.google.com	19
Figura 6.3 Campos del UDP en la solicitud.....	19
Figura 6.4 Campos del UDP en la respuesta.....	20

Lista de Tablas

Tabla 2.1 Conexión de cliente por Wifi con Univirtual	5
Tabla 2.2 Conexión de cliente por Wifi con Biblioteca	6
Tabla 2.3 Conexión de cliente por Ethernet con Web Local	6
Tabla 2.4 Información real enviada al servidor Web	9

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:

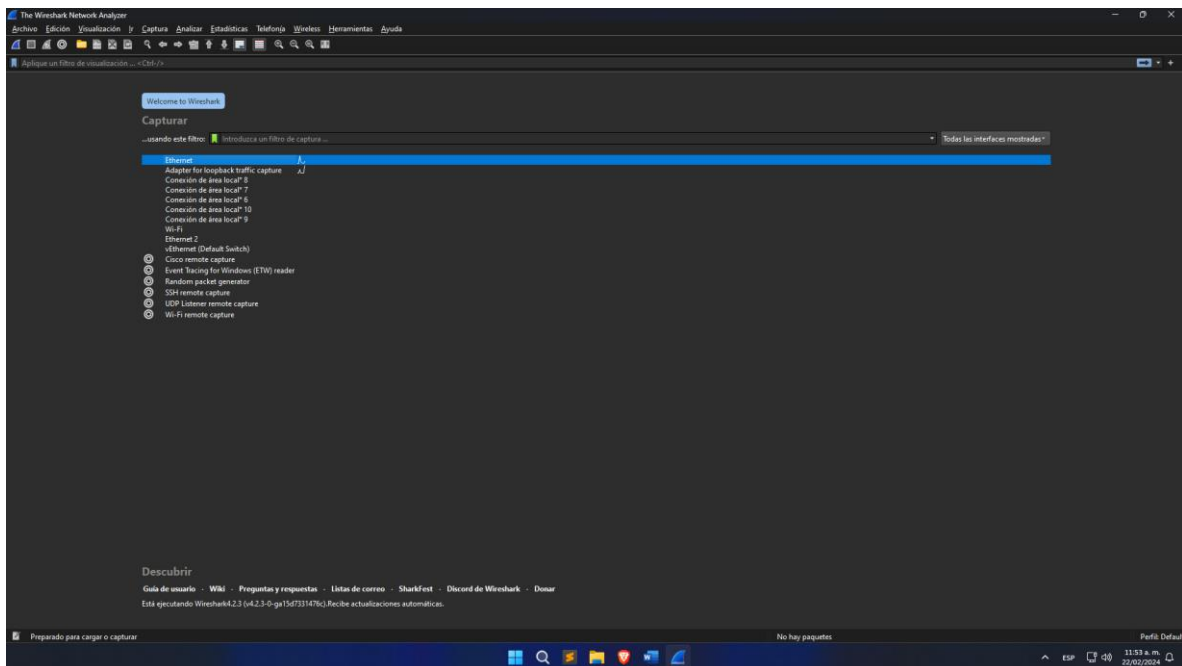


Figura 1.1 Interfaz del programa WireShark

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:

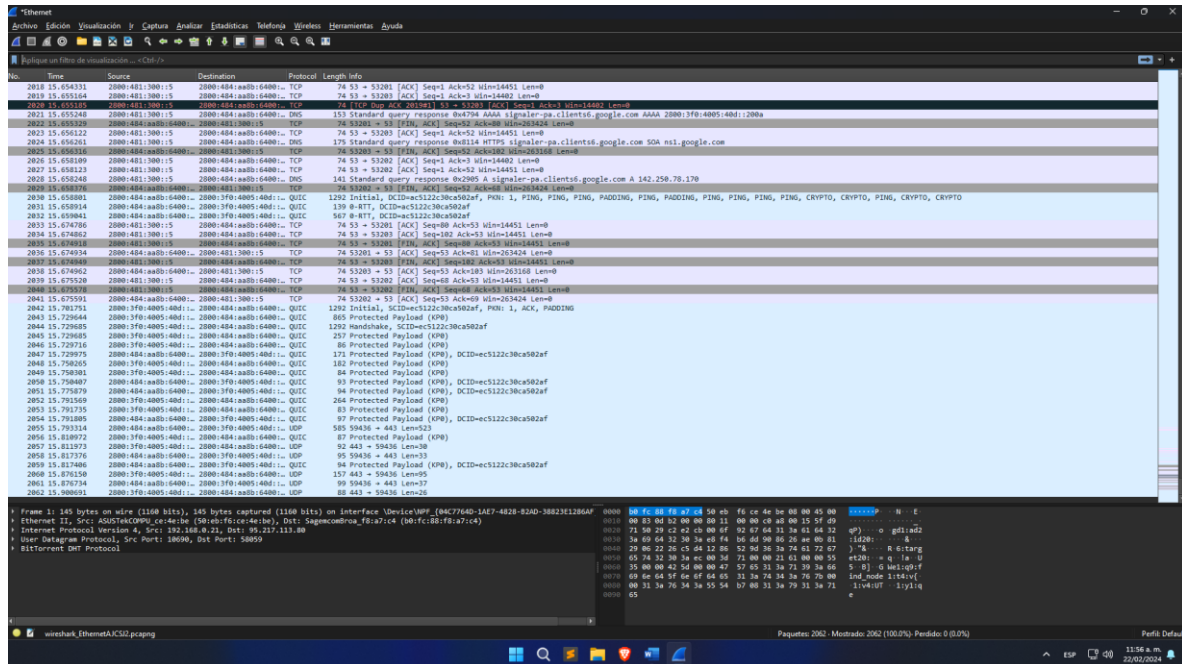


Figura 1.2 Registro de datos en Wireshark

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


```
Adaptador de Ethernet Ethernet:

Sufijo DNS específico para la conexión. . . : 
Vínculo: dirección IPv6 local. . . . . : fe80::ccae:ae51:5378:8f83%15
Dirección IPv4. . . . . : 192.168.128.15
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.128.2
```

Figura 2.1 Dirección IP cliente Ethernet

```
Adaptador de LAN inalámbrica Wi-Fi:

Sufijo DNS específico para la conexión. . . : unicauca.edu.co
Vínculo: dirección IPv6 local. . . . . : fe80::f821:506a:f293:e105%3
Dirección IPv4. . . . . : 10.132.27.3
Máscara de subred . . . . . : 255.255.254.0
Puerta de enlace predeterminada . . . . . : 10.132.26.1
```

Figura 2.2 Dirección IP cliente Wifi

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

```
Nombre: univirtual.unicauca.edu.co
Addresses: 2801:12:7000:3f4::43
           10.20.4.43
Aliases: www.univirtual.unicauca.edu.co
```

Figura 2.3 Dirección IP Servidor Univirtual

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

```
Nombre: biblio.unicauca.edu.co
Address: 10.20.5.11
```

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:

No.	Time	Source	Destination	Protocol	Length	Info
910	4.079146	10.132.27.3	10.20.4.43	TCP	66	60560 > https [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
920	4.079436	10.132.27.3	10.20.4.43	TCP	66	60561 > https [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
945	4.108143	10.20.4.43	10.132.27.3	TCP	66	https > 60560 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1386 SACK_PERM=1 WS=128
946	4.108226	10.132.27.3	10.20.4.43	TCP	54	60560 > https [ACK] Seq=1 Ack=1 Win=131584 Len=0
947	4.108614	10.132.27.3	10.20.4.43	TLSv1.2	792	Client Hello
949	4.108631	10.20.4.43	10.132.27.3	TCP	66	https > 60561 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1386 SACK_PERM=1 WS=128
953	4.108682	10.132.27.3	10.20.4.43	TCP	54	60561 > https [ACK] Seq=1 Ack=1 Win=131584 Len=0
954	4.109180	10.132.27.3	10.20.4.43	TLSv1.2	792	Client Hello
957	4.110910	10.132.27.3	10.20.4.43	TCP	66	60562 > http [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
958	4.111110	10.132.27.3	10.20.4.43	TCP	66	60563 > http [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
959	4.111365	10.20.4.43	10.132.27.3	TCP	56	https > 60560 [ACK] Seq=1 Ack=739 Win=30720 Len=0
960	4.113388	10.20.4.43	10.132.27.3	TCP	66	http > 60563 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1386 SACK_PERM=1 WS=128
961	4.113503	10.132.27.3	10.20.4.43	TCP	54	60563 > http [ACK] Seq=1 Ack=1 Win=131584 Len=0
963	4.123298	10.20.4.43	10.132.27.3	TCP	66	http > 60562 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1386 SACK_PERM=1 WS=128
965	4.123357	10.132.27.3	10.20.4.43	TCP	54	60562 > http [ACK] Seq=1 Ack=1 Win=131584 Len=0
966	4.124229	10.20.4.43	10.132.27.3	TLSv1.2	1440	Server Hello
967	4.124231	10.20.4.43	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
968	4.124232	10.20.4.43	10.132.27.3	TLSv1.2	1035	Certificate, Server Key Exchange, Server Hello Done
969	4.124281	10.132.27.3	10.20.4.43	TCP	54	60560 > https [ACK] Seq=739 Ack=3754 Win=131584 Len=0
973	4.125328	10.132.27.3	10.20.4.43	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Hello Request
975	4.127791	10.20.4.43	10.132.27.3	TLSv1.2	344	Encrypted Handshake Message, change cipher Spec, Encrypted Handshake Message
977	4.128138	10.132.27.3	10.20.4.43	TLSv1.2	1148	Application Data
978	4.131161	10.20.4.43	10.132.27.3	TCP	56	https > 60561 [ACK] Seq=1 Ack=739 Win=30720 Len=0
981	4.141926	10.20.4.43	10.132.27.3	TLSv1.2	1440	Server Hello
982	4.142881	10.20.4.43	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
983	4.142882	10.20.4.43	10.132.27.3	TLSv1.2	1035	Certificate, Server Key Exchange, Server Hello Done
984	4.142883	10.20.4.43	10.132.27.3	TLSv1.2	1034	Application Data, Application Data
985	4.142943	10.132.27.3	10.20.4.43	TCP	54	60561 > https [ACK] Seq=739 Ack=3754 Win=131584 Len=0
987	4.143867	10.132.27.3	10.20.4.43	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Hello Request
988	4.145668	10.20.4.43	10.132.27.3	TLSv1.2	344	Encrypted Handshake Message, change cipher Spec, Encrypted Handshake Message
990	4.147150	10.132.27.3	10.20.4.43	TLSv1.2	1163	Application Data
992	4.167221	10.20.4.43	10.132.27.3	TLSv1.2	1440	Application Data, Application Data
993	4.168476	10.20.4.43	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
994	4.168479	10.20.4.43	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
995	4.168479	10.20.4.43	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
996	4.168480	10.20.4.43	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]

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.

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

No.	Time	Source	Destination	Protocol	Length	Info
3507	12.335544	10.132.27.3	10.20.5.11	TCP	66	60565 > http [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3508	12.335874	10.132.27.3	10.20.5.11	TCP	66	60566 > http [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3511	12.337345	10.20.5.11	10.132.27.3	TCP	66	http > 60565 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1386 SACK_PERM=1 WS=32
3512	12.337414	10.132.27.3	10.20.5.11	TCP	54	60565 > http [ACK] Seq=1 Ack=1 Win=131584 Len=0
3514	12.337588	10.20.5.11	10.132.27.3	TCP	66	http > 60566 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1386 SACK_PERM=1 WS=32
3515	12.337633	10.132.27.3	10.20.5.11	TCP	54	60566 > http [ACK] Seq=1 Ack=1 Win=131584 Len=0
3517	12.339158	10.132.27.3	10.20.5.11	HTTP	690	GET / HTTP/1.1
3522	12.344226	10.20.5.11	10.132.27.3	TCP	56	http > 60565 [ACK] Seq=1 Ack=637 Win=15872 Len=0
3523	12.356420	10.20.5.11	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
3524	12.358218	10.20.5.11	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
3525	12.358219	10.20.5.11	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
3526	12.358220	10.20.5.11	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
3527	12.358221	10.20.5.11	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
3528	12.358222	10.20.5.11	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
3529	12.358222	10.20.5.11	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
3530	12.358223	10.20.5.11	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
3531	12.358224	10.20.5.11	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
3532	12.358225	10.20.5.11	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
3533	12.358291	10.132.27.3	10.20.5.11	TCP	54	60565 > http [ACK] Seq=637 Ack=13861 Win=131584 Len=0
3534	12.359821	10.20.5.11	10.132.27.3	TCP	1440	[TCP segment of a reassembled PDU]
3535	12.360308	10.20.5.11	10.132.27.3	HTTP	1294	HTTP/1.1 200 OK (text/html)
3536	12.360346	10.132.27.3	10.20.5.11	TCP	54	60565 > http [ACK] Seq=637 Ack=16487 Win=131584 Len=0
3833	13.534405	10.20.5.11	10.132.27.3	TCP	66	http > 60566 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1386 SACK_PERM=1 WS=32
3834	13.534405	10.132.27.3	10.20.5.11	TCP	66	60566 > http [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1386 SACK_PERM=1 WS=32
4609	17.361178	10.20.5.11	10.132.27.3	TCP	56	http > 60565 [FIN, ACK] Seq=16487 Ack=637 Win=15872 Len=0
4610	17.361252	10.132.27.3	10.20.5.11	TCP	54	60565 > http [ACK] Seq=637 Ack=16488 Win=131584 Len=0

Figura 2.6 Dirección IP servidor Biblioteca

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

No.	Time	Source	Destination	Protocol	Length	Info
7	3.252161	192.168.128.15	192.168.128.2	TCP	54	60516 > http [FIN, ACK] Seq=1 Ack=1 win=8212 Len=0
8	3.253561	192.168.128.2	192.168.128.15	TCP	60	http > 60516 [ACK] Seq=1 Ack=2 win=8212 Len=0
9	3.253561	192.168.128.2	192.168.128.15	TCP	60	http > 60516 [FIN, ACK] Seq=1 Ack=2 win=8212 Len=0
10	3.253605	192.168.128.15	192.168.128.2	TCP	54	60516 > http [ACK] Seq=2 Ack=2 win=8212 Len=0
23	12.598598	192.168.128.15	192.168.128.2	TCP	66	60529 > http [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
29	12.591972	192.168.128.15	192.168.128.2	TCP	66	60530 > http [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
30	12.598212	192.168.128.2	192.168.128.15	TCP	66	http > 60529 [SYN, ACK] Seq=0 Ack=1 win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
31	12.598262	192.168.128.15	192.168.128.2	TCP	54	60529 > http [ACK] Seq=1 Ack=1 win=262656 Len=0
32	12.598802	192.168.128.2	192.168.128.15	TCP	66	http > 60530 [SYN, ACK] Seq=0 Ack=1 win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
33	12.598847	192.168.128.15	192.168.128.2	TCP	54	60530 > http [ACK] Seq=1 Ack=1 win=2102272 Len=0
34	12.598965	192.168.128.15	192.168.128.2	HTTP	1223	GET /redes/public/login HTTP/1.1
35	12.651627	192.168.128.2	192.168.128.15	TCP	60	http > 60529 [ACK] Seq=1 Ack=1170 win=2102272 Len=0
36	12.969527	192.168.128.2	192.168.128.15	TCP	1514	[TCP segment of a reassembled PDU]
37	12.969528	192.168.128.2	192.168.128.15	TCP	1514	[TCP segment of a reassembled PDU]
38	12.969529	192.168.128.2	192.168.128.15	TCP	1514	[TCP segment of a reassembled PDU]
39	12.969535	192.168.128.2	192.168.128.15	TCP	1514	[TCP segment of a reassembled PDU]
40	12.969623	192.168.128.15	192.168.128.2	TCP	54	60529 > http [ACK] Seq=1170 Ack=5841 win=262656 Len=0
41	12.969875	192.168.128.2	192.168.128.15	TCP	1514	[TCP segment of a reassembled PDU]
42	12.969876	192.168.128.2	192.168.128.15	TCP	1514	[TCP segment of a reassembled PDU]
43	12.969877	192.168.128.2	192.168.128.15	TCP	1514	[TCP segment of a reassembled PDU]
44	12.969878	192.168.128.2	192.168.128.15	TCP	647	[TCP segment of a reassembled PDU]
45	12.969979	192.168.128.15	192.168.128.2	TCP	54	60529 > http [ACK] Seq=1170 Ack=10814 win=262656 Len=0
46	12.970416	192.168.128.2	192.168.128.15	HTTP	60	HTTP/1.1 200 OK (text/html)
47	13.013392	192.168.128.15	192.168.128.2	TCP	54	60529 > http [ACK] Seq=1170 Ack=10819 win=262656 Len=0
48	13.022218	192.168.128.15	192.168.128.2	HTTP	1137	GET /liverire/liverire.js?id=83b555bb3e243bc25f35 HTTP/1.1
49	13.024344	192.168.128.2	192.168.128.15	HTTP	595	HTTP/1.1 404 Not Found (text/html)
50	13.074227	192.168.128.15	192.168.128.2	TCP	54	60529 > http [ACK] Seq=2253 Ack=11360 win=262144 Len=0

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

- Complete la siguiente tabla con la información presentada en la sesión HTTP, hacer lo 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

- b. ¿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.

- c. 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:

No.	Time	Source	Destination	Protocol	Length	Info
919	4.079146	10.132.27.3	10.20.4.43	TCP	66	60560 > https [SYN] Seq=0 win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
920	4.079436	10.132.27.3	10.20.4.43	TCP	66	60561 > https [SYN] Seq=0 win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
945	4.108143	10.20.4.43	10.132.27.3	TCP	66	https > 60560 [SYN, ACK] Seq=0 Ack=1 win=29200 Len=0 MSS=1386 SACK_PERM=1 WS=128
946	4.108226	10.132.27.3	10.20.4.43	TCP	54	60560 > https [ACK] Seq=1 Ack=1 win=131584 Len=0
947	4.108614	10.132.27.3	10.20.4.43	TLSv1.2	792	client Hello
949	4.108631	10.20.4.43	10.132.27.3	TCP	66	https > 60561 [SYN, ACK] Seq=0 Ack=1 win=29200 Len=0 MSS=1386 SACK_PERM=1 WS=128
953	4.108682	10.132.27.3	10.20.4.43	TCP	54	60561 > https [ACK] Seq=1 Ack=1 win=131584 Len=0
954	4.109180	10.132.27.3	10.20.4.43	TLSv1.2	792	client Hello

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:

No.	Time	Source	Destination	Protocol	Length	Info
919	4.079146	10.132.27.3	10.20.4.43	TCP	66	60560 > https [SYN] Seq=0 win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
920	4.079436	10.132.27.3	10.20.4.43	TCP	66	60561 > https [SYN] Seq=0 win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
945	4.108143	10.20.4.43	10.132.27.3	TCP	66	https > 60560 [SYN, ACK] Seq=0 Ack=1 win=29200 Len=0 MSS=1386 SACK_PERM=1 WS=128
946	4.108226	10.132.27.3	10.20.4.43	TCP	54	60560 > https [ACK] Seq=1 Ack=1 win=131584 Len=0
947	4.108614	10.132.27.3	10.20.4.43	TLSv1.2	792	client Hello
949	4.108631	10.20.4.43	10.132.27.3	TCP	66	https > 60561 [SYN, ACK] Seq=0 Ack=1 win=29200 Len=0 MSS=1386 SACK_PERM=1 WS=128
953	4.108682	10.132.27.3	10.20.4.43	TCP	54	60561 > https [ACK] Seq=1 Ack=1 win=131584 Len=0
954	4.109180	10.132.27.3	10.20.4.43	TLSv1.2	792	client Hello

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:

No.	Time	Source	Destination	Protocol	Length	Info
919	4.079146	10.132.27.3	10.20.4.43	TCP	66	60560 > https [SYN] Seq=0 win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
920	4.079436	10.132.27.3	10.20.4.43	TCP	66	60561 > https [SYN] Seq=0 win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
945	4.108143	10.20.4.43	10.132.27.3	TCP	66	https > 60560 [SYN, ACK] Seq=0 Ack=1 win=29200 Len=0 MSS=1386 SACK_PERM=1 WS=128
946	4.108226	10.132.27.3	10.20.4.43	TCP	54	60560 > https [ACK] Seq=1 Ack=1 win=131584 Len=0
947	4.108614	10.132.27.3	10.20.4.43	TLSv1.2	792	client Hello
949	4.108631	10.20.4.43	10.132.27.3	TCP	66	https > 60561 [SYN, ACK] Seq=0 Ack=1 win=29200 Len=0 MSS=1386 SACK_PERM=1 WS=128
953	4.108682	10.132.27.3	10.20.4.43	TCP	54	60561 > https [ACK] Seq=1 Ack=1 win=131584 Len=0
954	4.109180	10.132.27.3	10.20.4.43	TLSv1.2	792	client Hello

Figura 2.10 Confirmación del cliente (ACK)

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

- d. ¿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:

3833	13.534405	10.20.5.11	10.132.27.3	TCP	66	http > 60566 [SYN, ACK] Seq=0 Ack=1 win=14600 Len=0 MSS=1386 SACK_PERM=1 WS=32
3834	13.534465	10.132.27.3	10.20.5.11	TCP	66	http > 60566 [ACK] Seq=16487 Ack=637 win=15872 Len=0 SLE=0 SRE=1
4609	17.361178	10.20.5.11	10.132.27.3	TCP	56	http > 60565 [FIN, ACK] Seq=16487 Ack=637 win=15872 Len=0
4610	17.361252	10.132.27.3	10.20.5.11	TCP	54	60565 > http [ACK] Seq=637 Ack=16488 win=131584 Len=0

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:

3833	13.534405	10.20.5.11	10.132.27.3	TCP	66	http > 60566 [SYN, ACK] Seq=0 Ack=1 win=14600 Len=0 MSS=1386 SACK_PERM=1 WS=32
3834	13.534465	10.132.27.3	10.20.5.11	TCP	66	http > 60566 [ACK] Seq=16487 Ack=637 win=15872 Len=0 SLE=0 SRE=1
4609	17.361178	10.20.5.11	10.132.27.3	TCP	56	http > 60565 [FIN, ACK] Seq=16487 Ack=637 win=15872 Len=0
4610	17.361252	10.132.27.3	10.20.5.11	TCP	54	60565 > http [ACK] Seq=637 Ack=16488 win=131584 Len=0

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.

- 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.

Wireshark - Paquete 71 - Ethernet

- Frame 71: 515 bytes on wire (4120 bits), 515 bytes captured (4120 bits) on interface \Device\NPF {04C7764D-1AE7-4828-B2AD-38823E1286AF}, id 0
- Ethernet II, Src: ASUSTekCOMPU ce:4e:be (50:eb:f6:ce:4e:be), Dst: SagemcomBroa_f8:a7:c4 (b0:fc:88:f8:a7:c4)
- Internet Protocol Version 4, Src: 192.168.0.21, Dst: 45.231.185.189
- Transmission Control Protocol, Src Port: 55363, Dst Port: 80, Seq: 1, Ack: 1, Len: 461
- Hypertext Transfer Protocol
 - GET / HTTP/1.1\r\n
 - Host: biblio.unicauca.edu.co\r\n
 - Connection: keep-alive\r\n
 - Upgrade-Insecure-Requests: 1\r\n
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36\r\n
 - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8\r\n
 - Sec-GPC: 1\r\n
 - Accept-Language: es-419,es;q=0.6\r\n
 - Accept-Encoding: gzip, deflate\r\n
 - If-Modified-Since: Fri, 23 Feb 2024 00:19:32 GMT\r\n
 - \r\n
 - [Full request URI: http://biblio.unicauca.edu.co/]
 - [HTTP request 1/1]
 - [Response in frame: 87]

Figura 2.13 Solicitud HTTP (GET)

- f. ¿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:

3534	12.359821	10.20.5.11	10.132.27.3	TCP	1440 (TCP segment of a reassembled PDU)
3535	12.360308	10.20.5.11	10.132.27.3	HTTP	1294 HTTP/1.1 200 OK (text/html)
3536	12.360346	10.132.27.3	10.20.5.11	TCP	54 60565 → http [ACK] Seq=637 Ack=16487 win=13

Figura 2.14 Trama de respuesta a solicitud GET

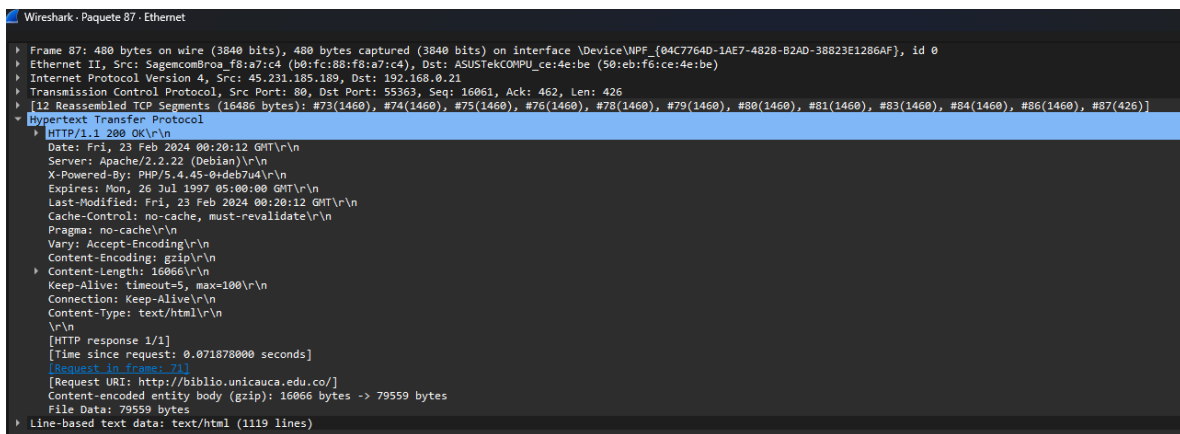


Figura 2.15 Datos encapsulados dentro del segmento TCP

- g. 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.

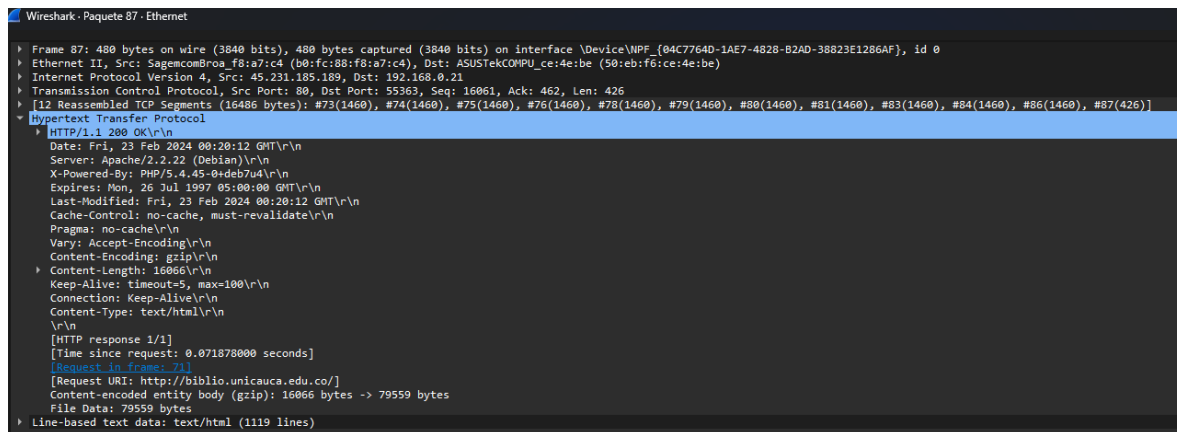
Versión del protocolo	HTTP/1.1
Método de solicitud	GET
Solicitud URI	/
Idioma	es-419, es;q=0.6

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.

- h. 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.



```
Wireshark - Paquete 87 - Ethernet
  Frame 87: 480 bytes on wire (3840 bits), 480 bytes captured (3840 bits) on interface \Device\NPF_{04C7764D-1AE7-482B-B2AD-38823E1286AF}, id 0
  Ethernet II, Src: SagemcomBroadcom_88:e3:c4 (d8:fc:88:f8:a7:c4), Dst: ASUSTekCOMPU_ce:4e:be (50:eb:f6:ce:4e:be)
  Internet Protocol Version 4, Src: 45.221.105.189, Dst: 192.168.0.21
  Transmission Control Protocol, Src Port: 80, Dst Port: 55363, Seq: 16061, Ack: 462, Len: 426
  [12 Reassembled TCP Segments (16486 bytes): #73(1460), #74(1460), #75(1460), #76(1460), #78(1460), #79(1460), #80(1460), #81(1460), #83(1460), #84(1460), #86(1460), #87(426)]
  Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Fri, 23 Feb 2024 00:20:12 GMT\r\n
    Server: Apache/2.2.22 (Debian)\r\n
    X-Powered-By: PHP/5.4.45-0+deb7u4\r\n
    Expires: Mon, 26 Jul 1997 05:00:00 GMT\r\n
    Last-Modified: Fri, 23 Feb 2024 00:20:12 GMT\r\n
    Cache-Control: no-cache, must-revalidate\r\n
    Pragma: no-cache\r\n
    Vary: Accept-Encoding\r\n
    Content-Encoding: gzip\r\n
    Content-Length: 16066\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.071878000 seconds]
    [Request URI: https://biblio.unicauca.edu.co/]
    Content-encoded entity body (gzip): 16066 bytes -> 79559 bytes
    File Data: 79559 bytes
  Line-based text data: text/html (1119 lines)
```

Figura 2.16 Respuesta del servidor detallada para la solicitud GET

- i. ¿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.


```

Hypertext Transfer Protocol
  HTTP/1.1 200 OK
  Date: Fri, 23 Feb 2024 00:28:12 GMT\r\n
  Server: Apache/2.2.22 (Debian)\r\n
  X-Powered-By: PHP/5.4.45\r\n
  Expires: Mon, 26 Jul 1997 05:00:00 GMT\r\n
  Last-Modified: Fri, 23 Feb 2024 00:20:12 GMT\r\n
  Cache-Control: no-cache, must-revalidate\r\n
  Pragma: no-cache\r\n
  Vary: Accept-Encoding\r\n
  Content-Encoding: gzip\r\n
  Content-Length: 10806\r\n
  Keep-Alive: timeout=5, max=100\r\n
  Connection: Keep-Alive\r\n
  Content-Type: text/html\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.0718000 seconds]
  [Request URI: http://biblio.unicauca.edu.co/]
  Content-encoded entity body (gzip): 10806 bytes -> 79559 bytes
  File Data: 79559 bytes
  + Line-based text data: text/html (1119 lines)
  \n
  <!DOCTYPE html>\n
  <html class="wide wow-animation" lang="en">\n
  <head>\n
  <!-- Site Title-->\n
  <title>División de Gestión Recursos Bibliográficos </title>\n
  <meta name="format-detection" content="telephone=no">\n
  <meta name="viewport" content="width=device-width, height=device-height, initial-scale=1.0, maximum-scale=1.0, user-scalable=0">\n
  <meta http-equiv="X-UA-Compatible" content="IE=edge">\n
  <meta charset="utf-8">\n
  <link rel="icon" href="images/favicon.png" type="image/x-icon">\n
  <!-- Stylesheets-->\n
  <link rel="stylesheet" type="text/css" href="//fonts.googleapis.com/css?family=Poppins:300,400,500,600,700,900">\n
  <link rel="stylesheet" href="css/bootstrap.css">\n
  <link rel="stylesheet" href="css/style.css">\n
  <link rel="stylesheet" href="css/fonts.css">\n
  <!--[if lt IE 10]>\n
  <script src="js/html5shiv.min.js"></script>\n
  <script src="js/html5shiv.min.js"></script>\n
  <!--[endif]>\n
  </head>\n
  <body>\n
  \n
  <div class="preloader">\n
  <div class="cssload-container">\n
  <div class="cssload-speeding-wheel"></div>\n
  </div>\n
  
```

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

j. ¿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.

- k. 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.

```

If-Modified-Since: Mon, 07 Oct 2013 09:40:42 GMT\r\n
If-None-Match: "60a70-130d-4e8237219a9d6"\r\n

```

Localice esta información en Wireshark y dejar la respectiva evidencia.

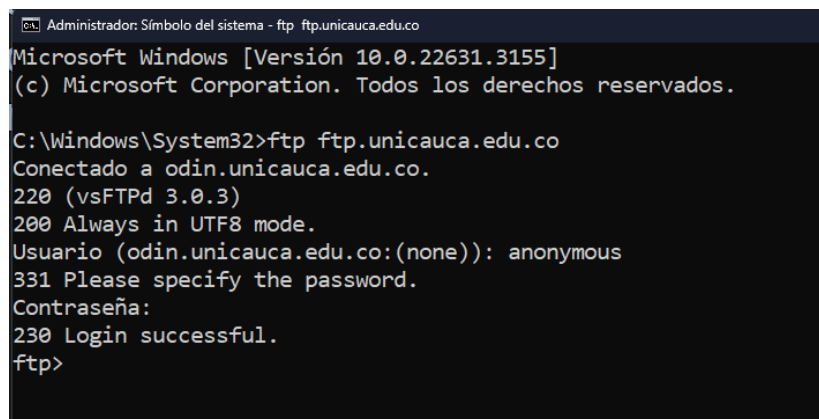
3.- Actividad 3: Captura y análisis de la comunicación FTP entre la computadora host del módulo y un servidor Web

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

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

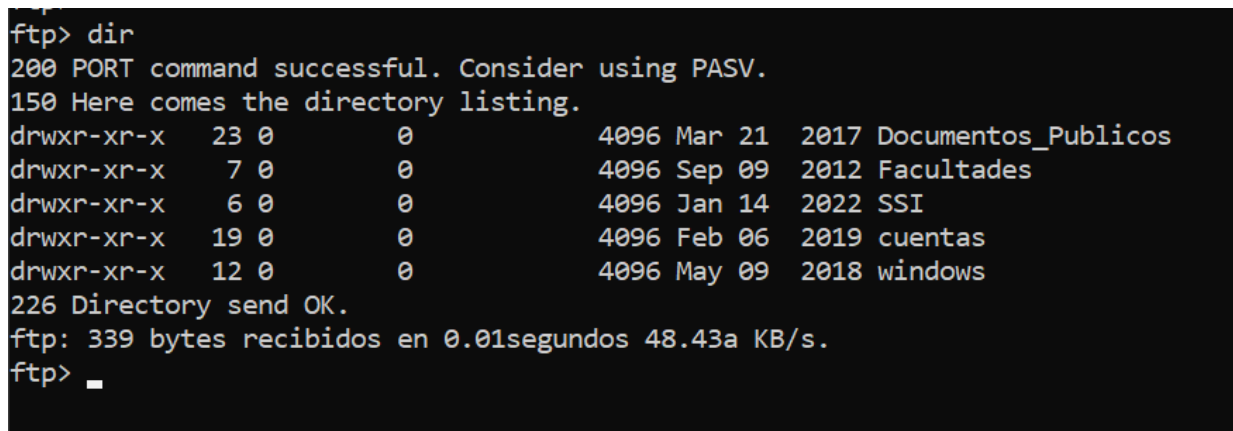


```
Administrador: Símbolo del sistema - ftp ftp.unicauca.edu.co
Microsoft Windows [Versión 10.0.22631.3155]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\System32>ftp ftp.unicauca.edu.co
Conectado a odin.unicauca.edu.co.
220 (vsFTPD 3.0.3)
200 Always in UTF8 mode.
Usuario (odin.unicauca.edu.co:(none)): anonymous
331 Please specify the password.
Contraseña:
230 Login successful.
ftp>
```

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

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



```
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x  23 0      0      4096 Mar 21  2017 Documentos_Publicos
drwxr-xr-x   7 0      0      4096 Sep 09  2012 Facultades
drwxr-xr-x   6 0      0      4096 Jan 14  2022 SSI
drwxr-xr-x  19 0      0      4096 Feb 06  2019 cuentas
drwxr-xr-x  12 0      0      4096 May 09  2018 windows
226 Directory send OK.
ftp: 339 bytes recibidos en 0.01segundos 48.43a KB/s.
ftp> _
```

Figura 3.2 Contenido del directorio actual

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

```
ftp> cd windows/Documentos/
250 Directory successfully changed.
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rwxr-xr-x  1 0      0      1819369 Oct 25  2013 Liberar_espacio_Outlook.pdf
-rw-r--r--  1 0      0      559576 Sep 26  2013 Liberar_espacio_Outlook_express.pdf
-rwxr-xr-x  1 0      0      1838214 Oct 25  2013 Manual_liberacion_espacio_Outlook_2007.pdf
-rwxr-xr-x  1 0      0      1819369 Oct 25  2013 Manual_liberacion_espacio_Outlook_2013.pdf
226 Directory send OK.
ftp: 381 bytes recibidos en 0.01segundos 76.20a KB/s.
ftp>
```

Figura 3.3 Accediendo a windows/Documentos

```
ftp> get Manual_liberacion_espacio_Outlook_2013.pdf
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for Manual_liberacion_espacio_Outlook_2013.pdf (1819369 bytes).
226 Transfer complete.
ftp: 1819369 bytes recibidos en 0.41segundos 4384.02a KB/s.
```

Figura 3.4 Descargando un archivo

7. Se cierra la ventana de línea de comandos.
8. 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/>.

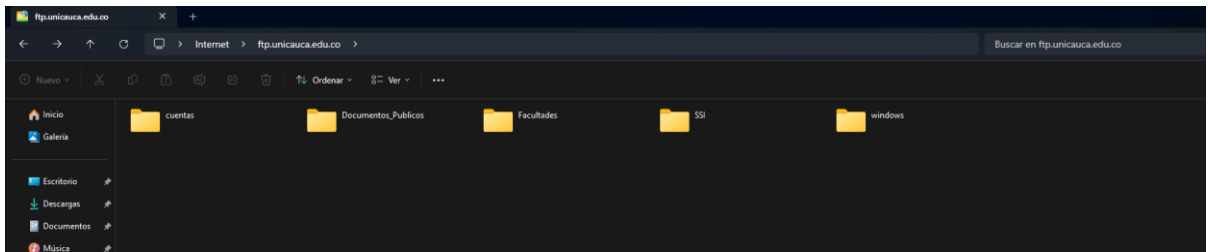


Figura 3.6 Acceso a ftp://ftp.unicauca.edu.co/ desde el explorador de archivos

3. Utilizando el explorador se procede a entrar a windows/Documentos y se guarda el archivo Liberar_espacio_Outlook.pdf.

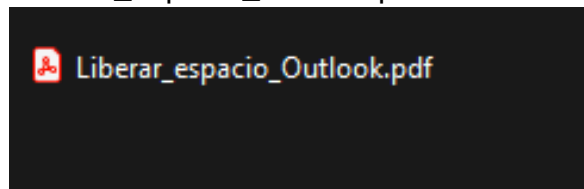


Figura 3.7 Archivo descargado

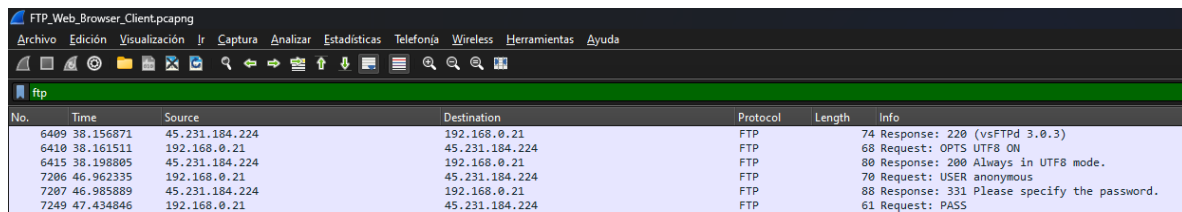
4. Se cierra el explorador.
5. Se detienen las capturas Wireshark y se guarda como FTP_Web_Browser_Client.



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.



No.	Time	Source	Destination	Protocol	Length	Info
6409	38.156871	45.231.184.224	192.168.0.21	FTP	74	Response: 220 (vsFTPd 3.0.3)
6410	38.161511	192.168.0.21	45.231.184.224	FTP	68	Request: OPTS UTF8 ON
6415	38.198805	45.231.184.224	192.168.0.21	FTP	80	Response: 200 Always in UTF8 mode.
7206	46.962335	192.168.0.21	45.231.184.224	FTP	70	Request: USER anonymous
7207	46.985889	45.231.184.224	192.168.0.21	FTP	88	Response: 331 Please specify the password.
7249	47.434846	192.168.0.21	45.231.184.224	FTP	61	Request: PASS

Figura 3.9 Captura FTP de respuesta 220

3. Se expande el protocolo FTP.

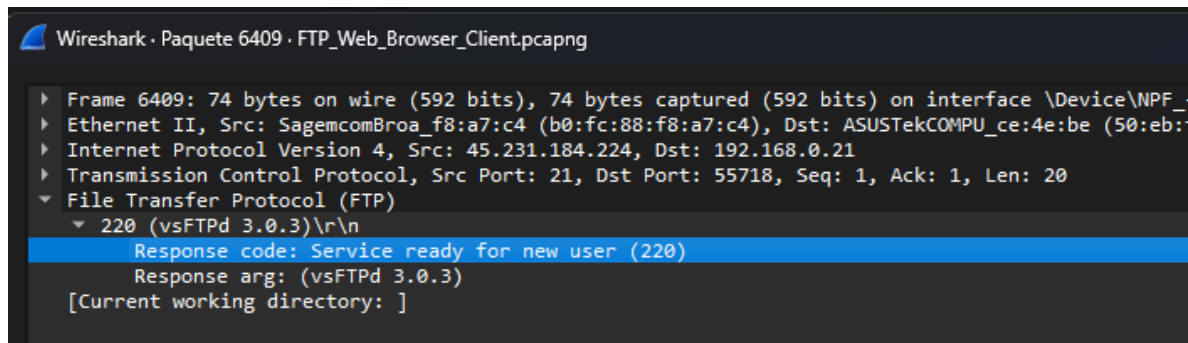
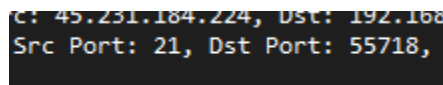


Figura 3.10 Protocolo FTP

- ¿Cuál es la respuesta 220 del servidor FTP?
Indica que el servicio está listo para un nuevo usuario.
- 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.
- ¿Qué número de puerto utiliza el cliente FTP para conectarse al puerto 21 del servidor FTP?
Utiliza el puerto 55718
- ¿Cuál es el número de puerto de Datos FTP utilizado por el servidor FTP?
Utiliza el número de puerto 21



```

Src Port: 21, Dst Port: 55718,
    
```

Figura 3.11 Puertos de cliente y servidor FTP

4. Se abre la captura FTP_Command_Line_Client y se comparan los clientes por línea de comando y browser.

The figure displays two side-by-side Wireshark packet capture windows. The left window, titled 'FTP_Command_Line_Client.pcapng', shows a list of packets with the first packet selected. The details pane on the right shows the FTP protocol structure, including the 'FTP-Data' field and the 'FTP-Data' field. The right window, titled 'FTP_Web_Browser_Client.pcapng', also shows a list of packets with the first packet selected. The details pane on the right shows the FTP protocol structure, including the 'FTP-Data' field and the 'FTP-Data' field. Both windows show a list of packets with the first packet selected, and the details pane on the right shows the FTP protocol structure, including the 'FTP-Data' field and the 'FTP-Data' field.

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:

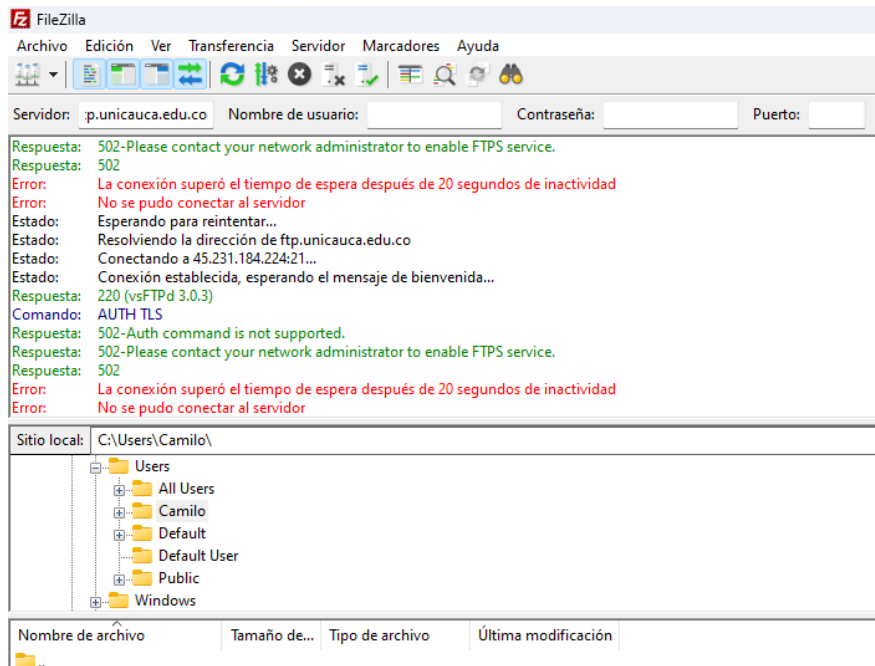


Figura 5.1 Intento de conexión al servidor FTP de la Universidad del Cauca

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

1. Se habilita la captura en Wireshark y se accede a www.google.com en el explorador Web.
2. Se detiene la captura en Wireshark.
3. Se filtra por dns.

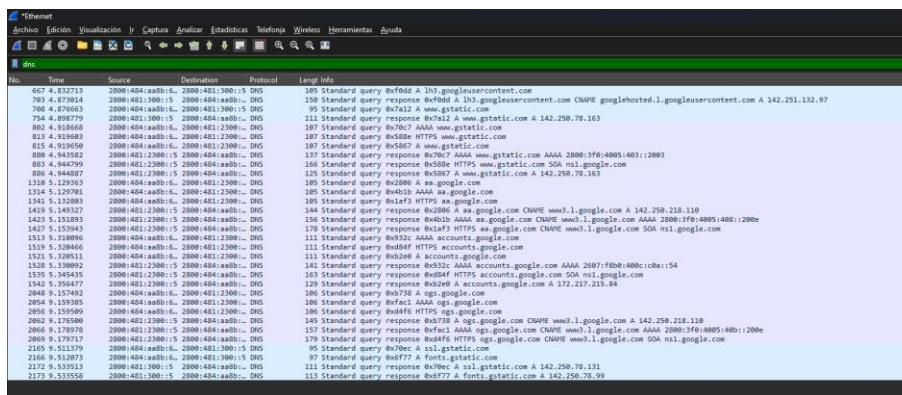


Figura 6.1 Filtro por dns en Wireshark

4. Se ubica el paquete standard query y "A www.google.com".

5. Se examina el UDP capturado por Wireshark.

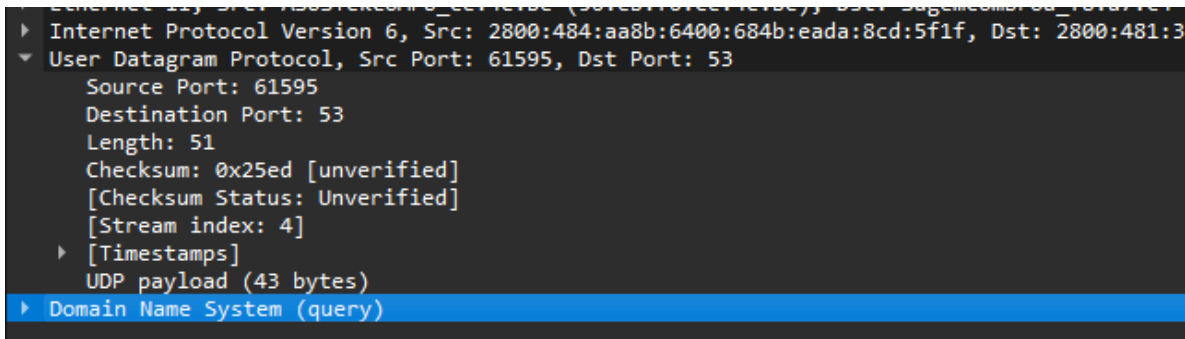


Figura 6.2 UDP por dns en Wireshark para www.google.com

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

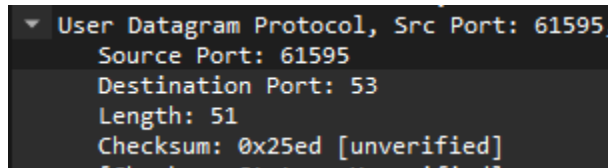


Figura 6.3 Campos del UDP en la solicitud

- 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.
7. 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.
8. 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:

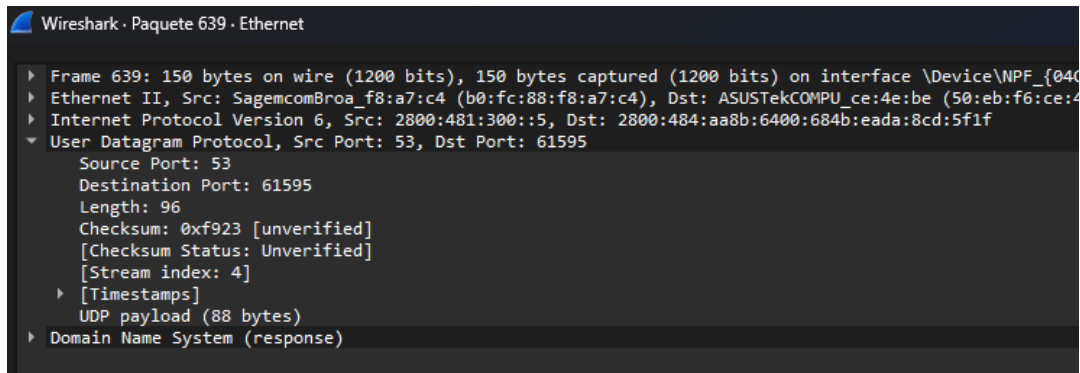


Figura 6.4 Campos del UDP en la respuesta

- ¿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.
9. ¿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.

10. ¿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>
- 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