

Laboratorio #1 - Red Humana e Intro a Wireshark

Descripción

Esta práctica está diseñada para analizar y comprender los fundamentos de la comunicación de datos, explorando las ventajas y desventajas de diferentes esquemas de transmisión. Se identificará la complejidad inherente al envío de información, además de familiarizarse con el funcionamiento básico de un conmutador a pequeña escala. Además, se busca familiarizarse con el entorno Wireshark para fortalecer la teoría sobre paquetes a través del análisis de paquetes reales.

Primera parte

Primera parte: transmisión de códigos

- ¿Qué esquema es más fácil? ¿Más difícil?
El código Morse fue el esquema de transmisión de mensajes más fácil de usar debido a su lógica intuitiva. En contraste, el código Baudot presentó una mayor dificultad por la complejidad de su estructura.
- ¿Con cuál ocurren menos errores?
El código Morse era más fácil de reconocer y aprender, lo que lo hacía más accesible su interpretación. Asimismo, fue más fácil transmitirlo por medio de voz; por lo que, se cometieron menos errores.

Segunda parte: transmisión “empaquetada”

- ¿Qué dificultades involucra el enviar un mensaje de esta forma “empaquetada”?
Si el emisor cometía un error en la transmisión, el receptor recibía un mensaje incorrecto (sin interrupción, aunque se detectara el fallo). Además, el emisor debía asegurar un entorno libre de ruidos que pudieran interferir con la grabación. Por lo tanto, la mayor dificultad radicaba en controlar el entorno, dado que el ritmo y las pausas son importantes.

Tercera parte: conmutación de mensajes

- ¿Qué posibilidades incluye la introducción de un conmutador en el sistema?
Al introducir un conmutador en el sistema ayuda a centralizar la comunicación, es decir que puede organizar como se llevan a cabo la transmisión de los mensajes. Además, es como un filtro para priorizar mensajes y tener una estructura.
- ¿Qué ventajas/desventajas se tienen al momento de agregar más conmutadores al sistema?
Ventajas: si se tiene muchos mensajes, tener más de un conmutador puede ayudar al flujo de recepción y envío. También, ayuda a la tolerancia de fallos, ya que si un conmutador falla el otro puede tomar su lugar.

Gabriela de León 21037 / Lourdes Saavedra 21333
Sofia Lam 21548 / Alexander Cuxé 22648

Desventajas: la red se vuelve más compleja al aumentar la cantidad de conmutadores debido a que se debe de agregar un orden y reglas entre conmutadores. Asimismo, es posible que el tiempo entre la recepción y el envío del mensaje aumente.

Detalle del protocolo:

- Durante esta parte diseñamos un protocolo de comunicación sencillo que permitiera organizar el envío de mensajes entre clientes sin generar confusión ni sobrecargar al intermediario. Para indicar el destino del mensaje, se usó un encabezado claro en las notas de voz, como "Destino: B. Mensaje: ...". Esta estructura permitió al conmutador identificar con rapidez a quién debía reenviar el mensaje, transmitiendo únicamente el contenido codificado. También se establecieron reglas para el control del flujo: los mensajes se enviaban por turnos, con confirmaciones de disponibilidad por parte del conmutador, y se limitó a dos mensajes consecutivos por cliente para equilibrar la carga.

Además, se contemplaron medidas para el manejo de errores y congestión. En caso de múltiples envíos simultáneos, los mensajes se procesaban en orden de llegada y se notificaba a los clientes que esperaran. Si el mensaje era confuso o incompleto, el conmutador pedía una repetición, y una vez reenviado, el receptor confirmaba la recepción, cerrando así el ciclo. Este ejercicio resultó muy útil para comprender cómo los protocolos reales implementan mecanismos similares de direccionamiento, control de tráfico y gestión de errores.

Segunda parte

Primera parte: personalización del entorno

The screenshot displays the Wireshark interface with a list of captured packets on the left and a detailed view of a selected packet on the right. The packet list shows various protocols including TCP, TLSv1.3, and Application. The detailed view on the right shows the raw packet data in hexadecimal and ASCII, along with a packet bytes overview.

No.	Time	Source	Destination	Protocol	TCP Segment Len	Info
1	11:16:47.126585	192.168.0.46	172.67.75.39	TCP	0	5111
2	11:16:47.177831	172.67.75.39	192.168.0.46	TCP	0	443
3	11:16:47.177959	192.168.0.46	172.67.75.39	TCP	0	5111
4	11:16:47.178243	192.168.0.46	172.67.75.39	TLSv1.3		517 C14
5	11:16:47.255959	172.67.75.39	192.168.0.46	TCP	0	443
6	11:16:47.267905	172.67.75.39	192.168.0.46	TLSv1.3		1460 Ser
7	11:16:47.267905	172.67.75.39	192.168.0.46	TLSv1.3		348 App
8	11:16:47.267999	192.168.0.46	172.67.75.39	TCP	0	5111
9	11:16:49.048600	192.168.0.46	172.67.75.39	TLSv1.3		64 Cha
10	11:16:49.048918	192.168.0.46	172.67.75.39	TLSv1.3		92 App
11	11:16:49.049309	192.168.0.46	172.67.75.39	TLSv1.3		666 App
12	11:16:49.073867	172.67.75.39	192.168.0.46	TCP	0	443
13	11:16:49.073867	172.67.75.39	192.168.0.46	TLSv1.3		528 App
14	11:16:49.074157	192.168.0.46	172.67.75.39	TLSv1.3		31 App
15	11:16:49.079528	172.67.75.39	192.168.0.46	TCP	0	443
16	11:16:49.079528	172.67.75.39	192.168.0.46	TCP	0	443
17	11:16:49.079528	172.67.75.39	192.168.0.46	TLSv1.3		31 App
18	11:16:49.099770	172.67.75.39	192.168.0.46	TCP	0	443
19	11:16:49.123997	192.168.0.46	172.67.75.39	TCP	0	5111
20	11:16:49.389360	172.67.75.39	192.168.0.46	TLSv1.3		1391 App
21	11:16:49.389360	172.67.75.39	192.168.0.46	TLSv1.3		1460 App
22	11:16:49.389360	172.67.75.39	192.168.0.46	TLSv1.3		1322 App
23	11:16:49.389360	172.67.75.39	192.168.0.46	TLSv1.3		1391 App
24	11:16:49.389360	172.67.75.39	192.168.0.46	TLSv1.3		1460 App
25	11:16:49.389360	172.67.75.39	192.168.0.46	TLSv1.3		1322 App
26	11:16:49.389360	172.67.75.39	192.168.0.46	TLSv1.3		157 App
27	11:16:49.389546	192.168.0.46	172.67.75.39	TCP	0	5111
28	11:16:49.461370	172.67.75.39	192.168.0.46	TLSv1.3		1391 App
29	11:16:49.464847	172.67.75.39	192.168.0.46	TLSv1.3		1391 App
30	11:16:49.464847	172.67.75.39	192.168.0.46	TLSv1.3		1460 App
31	11:16:49.464847	172.67.75.39	192.168.0.46	TLSv1.3		1322 App
32	11:16:49.464847	172.67.75.39	192.168.0.46	TLSv1.3		1460 App
33	11:16:49.464847	172.67.75.39	192.168.0.46	TLSv1.3		1460 App
34	11:16:49.464847	172.67.75.39	192.168.0.46	TLSv1.3		1460 App
35	11:16:49.464847	172.67.75.39	192.168.0.46	TLSv1.3		1184 App
36	11:16:49.464847	172.67.75.39	192.168.0.46	TLSv1.3		1460 App
37	11:16:49.464847	172.67.75.39	192.168.0.46	TLSv1.3		1322 App
38	11:16:49.464847	172.67.75.39	192.168.0.46	TLSv1.3		1460 App
39	11:16:49.464847	172.67.75.39	192.168.0.46	TLSv1.3		1460 App
40	11:16:49.464847	172.67.75.39	192.168.0.46	TLSv1.3		1460 App
41	11:16:49.464847	172.67.75.39	192.168.0.46	TLSv1.3		1460 App

Gabriela de León 21037 / Lourdes Saavedra 21333
Sofia Lam 21548 / Alexander Cuxé 22648

Segunda parte: configuración de la captura de paquetes

-

```
PS C:\Users\gabid> ipconfig

Windows IP Configuration

Wireless LAN adapter Conexión de área local* 1:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Conexión de área local* 2:

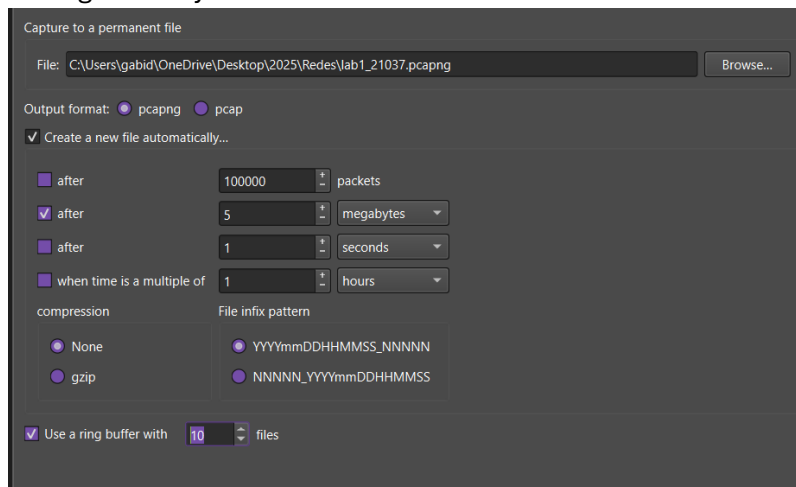
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::b0cf:1c6d:4155:dada%13
    IPv4 Address. . . . . : 192.168.83.250
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.83.44
PS C:\Users\gabid> |
```

Si se ejecuta ipconfig en la terminal se puede observar la configuración de red de las interfaces. Aquí se observa dos adaptadores inalámbricos y un adaptador de Wi-Fi con conexión estable. Además, la interfaz de red activa se llama Wireless LAN adapter Wi-Fi.

- Configuración y comandos



Gabriela de León 21037 / Lourdes Saavedra 21333
Sofia Lam 21548 / Alexander Cuxé 22648

Name	Status	Date modified	Type	Size
lab1_21037_20250717183341_00014	●	7/17/2025 6:33 PM	File	4,884 KB
lab1_21037_20250717183346_00015	●	7/17/2025 6:33 PM	File	4,885 KB
lab1_21037_20250717183347_00016	●	7/17/2025 6:33 PM	File	4,884 KB
lab1_21037_20250717183350_00017	●	7/17/2025 6:33 PM	File	4,883 KB
lab1_21037_20250717183352_00018	●	7/17/2025 6:33 PM	File	4,885 KB
lab1_21037_20250717183352_00019	●	7/17/2025 6:33 PM	File	4,884 KB
lab1_21037_20250717183357_00020	●	7/17/2025 6:33 PM	File	4,885 KB
lab1_21037_20250717183358_00021	●	7/17/2025 6:33 PM	File	4,885 KB
lab1_21037_20250717183359_00022	●	7/17/2025 6:34 PM	File	4,884 KB
lab1_21037_20250717183401_00023	●	7/17/2025 6:34 PM	File	3,234 KB

Tercera parte: análisis de paquetes

No.	Time	Source	Dest	Protocol	Length	Info
3393	19:12:04.164490	192.168.83.250	128.119.245.12	HTTP	574	GET / HTTP/1.1
3421	19:12:04.279878	128.119.245.12	192.168.83.250	HTTP	574	200 OK

Frame 3353: 574 bytes on wire (4592 bits), 574 bytes captured (4592 bits) on interface 0
Ethernet II, Src: AzureWaveTec_bd:25:dd (28:d0:43:bd:25:dd), Dst: 192.168.83.250
Internet Protocol Version 4, Src: 192.168.83.250, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 61987, Dst Port: 80, Seq: 3393, Len: 574
Hypertext Transfer Protocol

0000	e2 5c 0c ad de 22 28 d0 43 bd 25 dd 08 00 41
0010	02 30 42 f0 40 00 80 06 2b b1 c0 a8 53 fa 81
0020	f5 0c f2 23 00 50 6a bd b4 bd 3f ad e1 6d 51
0030	00 ff 78 96 00 00 47 45 54 20 2f 77 69 72 6e
0040	68 61 72 6b 2d 6c 61 62 73 2f 49 4e 54 52 41
0050	77 69 72 65 73 68 61 72 6b 2d 66 69 6c 65 33
0060	68 74 6d 6c 20 48 54 54 50 2f 31 2e 31 0d 00
0070	6f 73 74 3a 20 67 61 69 61 2e 63 73 2e 75 6e
0080	73 73 2e 65 64 75 0d 0a 43 6f 6e 6e 65 63 71
0090	6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 65 00
00a0	55 70 67 72 61 64 65 2d 49 6e 73 65 63 75 71
00b0	2d 52 65 71 75 65 73 74 73 3a 20 31 0d 0a 51
00c0	65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6e
00d0	61 2f 35 2e 30 20 28 57 69 6e 64 6f 77 73 21
00e0	54 20 31 30 2e 30 3b 20 57 69 6e 36 34 3b 21
00f0	36 34 29 20 41 70 70 6c 65 57 65 62 4b 69 71
0100	35 33 37 2e 33 36 20 28 4b 48 54 4d 4c 2c 21
0110	69 6b 65 20 47 65 63 6b 6f 29 20 43 68 72 6e
0120	65 2f 31 33 38 2e 30 2e 30 2e 30 20 53 61 6e
0130	72 69 2f 35 33 37 2e 33 36 20 45 64 67 2f 31
0140	38 2e 30 2e 30 2e 30 0d 0a 41 63 63 65 70 71
0150	20 74 65 70 74 2f 69 74 64 6e 71 61 70 70 6e

Hypertext Transfer Protocol: Protocol Packets: 4044 · Displayed: 4 (0.1%) · Dropped: 0 (0.0%) Profile: Gabriela de León

- a) ¿Qué versión de HTTP está ejecutando su navegador?
La versión que utiliza es HTTP/1.1.
- b) ¿Qué version de HTTP está ejecutando el servidor?
La versión que utiliza es HTTP/1.1.

- c) ¿Qué lenguajes (si aplica) indica el navegador que acepta a el servidor?

El lenguaje que acepta es:

```
Accept-Language: es,es-ES;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n
```

- d) ¿Cuántos bytes de contenido fueron devueltos por el servidor?

El servidor puede devolver 81.

```
Content-Length: 81\r\n
```

- e) En el caso que haya un problema de rendimiento mientras se descarga la página, ¿en que dispositivos de la red convendría “escuchar” los paquetes? ¿Es conveniente instalar Wireshark en el servidor? Justifique

En los dispositivos que se debería “escuchar” sería al usuario y el punto de acceso. Esto para evaluar cómo es enviada y recibida la información. Por otro lado, usar Wireshark como el servidor no es conveniente debido a que solo se puede visualizar desde la perspectiva del servidor.

Discusión

El primer parte del laboratorio permitió una comprensión práctica de cómo cambia la transmisión de información y sus complejidades. Se observó cómo la introducción de un conmutador impacta la transmisión de datos, incluyendo sus ventajas y desventajas inherentes. Por otra parte, el entorno de Wireshark es una herramienta para visualizar el tráfico de red en tiempo real, permitiendo la identificación y el diagnóstico de posibles problemas dentro de la red. En ello se puede analizar los paquetes de la red para identificar como se encuentra la red.

La experiencia con Wireshark resultó complicada, ya que la información presentada era completamente nueva para mí. La dificultad se debió al desconocimiento de la terminología específica de la herramienta, la ubicación de datos relevantes y cómo interpretar la información mostrada. Por otro lado, la interacción con los códigos Morse y Baudot fue divertida, aunque también generó cierta confusión inicial al intentar comprender su funcionamiento.

Comentarios

Me gustaría mencionar que deseo que las instrucciones para el uso de Wireshark fueran más específicas (se podría incluir imágenes), ya que eran algo confusas de como llevarlas a cabo.

Conclusiones

- La simulación para la conmutación de mensajes como red humana muestra las dificultades del envío de información.

Gabriela de León 21037 / Lourdes Saavedra 21333

Sofia Lam 21548 / Alexander Cuxé 22648

- Wireshark es una herramienta de visualización del tráfico de red ayudó a comprender como se compone la solicitud HTTP.
- Wireshark ayudó comprender como es el funcionamiento de los protocolos.

Referencias

Wireshark Foundation. (s.f.). Wireshark® documentation. Wireshark Foundation.
<https://www.wireshark.org/docs/>