Laboratorio 1: Red Humana e Intro a Wireshark

Nombre y carnet / Nombre y carne de pareja

Mi nombre: Lourdes Saavedra – 21333

Mi pareja: Gabriela De León – 21037

Nombres y carnet de la otra pareja

Sofía Lam – 21548

Alexander Cuxé – 22648

Título de la práctica

Red Humana e Intro a Wireshark

Descripción de la práctica

La práctica se divide en dos bloques: una simulación de red humana mediante esquemas de codificación y una introducción al uso de Wireshark. En la primera parte se experimenta con dos esquemas de codificación de información: el código Morse y el código Baudot. Con ayuda de una pareja se envían y reciben mensajes utilizando sonidos generados con la voz o con algún objeto para simular los dos esquemas de codificación. Mediante este ejercicio se busca observar las dificultades involucradas en la transmisión precisa de información sin un medio digital. Además, se busca identificar la posibilidad de errores, la importancia de los tiempos y eficiencia de los distintos códigos.

Luego de esto se repite la dinámica utilizando medios como mensajes de voz WhatsApp o Discord. Esto permite incorporar la idea de "empaquetamiento" en la transmisión de datos. Esta parte nos permite darnos cuenta de los retos de la comunicación no en tiempo real y la necesidad de protocolos y estructuras claras. Finalmente, en la tercera fase del ejercicio grupal se simula el funcionamiento de una red conmutada. Para ello se introdujo el rol de un conmutador humano que intermedia entre los clientes para dirigir los mensajes a sus destinos correctos. Esta actividad permite entender mejor conceptos como direccionamiento, control de tráfico y escalabilidad de redes.

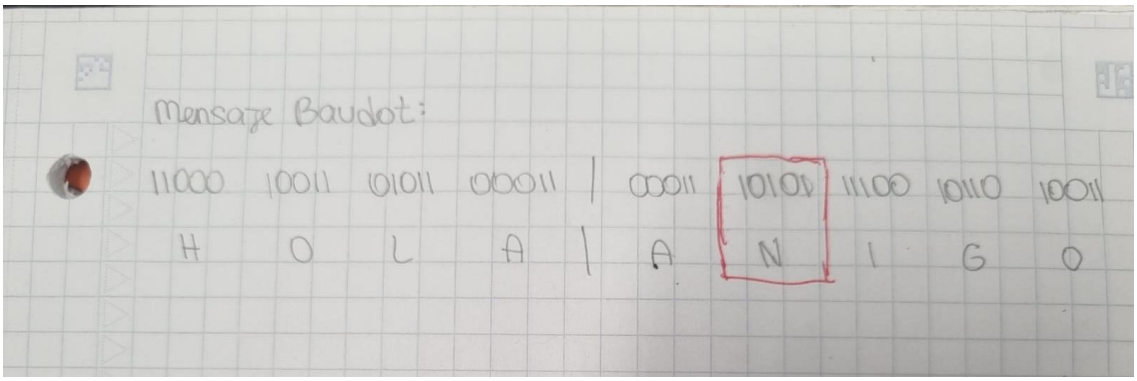
Ahora, en la segunda parte de la práctica se introduce la herramienta Wireshark. En primer lugar se personaliza el entorno de trabajo, aprendiendo a crear perfiles, aplicar filtros, reglas de color y configurar la interfaz para un análisis eficiente. Luego se realiza una captura de paquetes de red con configuración de ring buffer, explorando comandos como ipconfig o ifconfig y reconociendo las distintas interfaces de red disponibles. Finalmente, se analizan capturas reales de tráfico HTTP, identificando versiones de protocolo, cabeceras, volúmenes de datos y posibles cuellos de botella en la red.

Respuestas de las preguntas de la primera parte

- ¿Qué esquema (código) fue más fácil de transmitir y por qué? ¿Qué esquema (código) fue más difícil de transmitir y por qué?
  El esquema más fácil de usar fue el Morse porque tiene una lógica más intuitiva, con sonidos que son fáciles de reconocer. Además, se pueden memorizar un poco más fácil. Mientras que el de Baudot usa combinaciones más abstractas y se puede llegar a complicar un poco más.

- ¿Qué esquema tuvo menos errores (incluir datos que lo evidencien)?
  Observamos que se cometen menos errores con el esquema más simple e intuitivo que es el Morse. Esto porque se puede reconocer más fácil cuando se termina una letra o palabra. Además, están las señales auditivas más distinguibles. En cambio en el otro esquema, se trataba de secuencias exacta, que podían llegar a ser similares, por lo que hubo confusiones, omisiones o errores de interpretación.

  Evidencia:



  En uno de los mensajes transmitidos con Baudot ("HOLA AMIGO"), la persona que recibió el mensaje entendió "HOLA ANIGO". Esto porque la secuencia de

bits para la letra "M" fue mal interpretada como "N". Esto pudo ser por un error de ritmo, confusión en la transmisión o como consecuencia por estar en zoom. Este tipo de confusiones fue recurrente debido a lo similares que pueden ser algunas secuencias. En cambio, todos los mensajes transmitidos en código Morse fueron comprendidos correctamente, ya que las pausas entre letras y palabras, así como la duración diferenciada de los sonidos (cortos y largos), facilitaron su interpretación.

- ¿Qué dificultades involucra el enviar un mensaje de forma “empaquetada”?
  En primer lugar se tiene la falta de retroalimentación inmediata pues no hay confirmación o corrección en ese momento. Entonces si hubo un error, no se puede aclarar en ese momento. También se encuentra el hecho de que el receptor depende de la claridad del audio. Finalmente puede haber un retraso en la comunicación, ya que transcurre tiempo entre el envió, la escucha y la respuesta.

- ¿Qué ventajas/desventajas se tienen al momento de agregar más conmutadores al sistema?
  Dentro de las ventajas se encuentra que se evita la sobrecarga de uno si hay muchos mensajes. Además, en caso de una falla en alguno, otro puede seguir funcionando. También puede haber una repartición de tareas, donde diferentes conmutadores puedan cumplir funciones específicas.

  Dentro de las desventajas se puede mencionar que hay una mayor complejidad. Además, puede haber un retraso en la entrega del mensaje porque este tiene que pasar por varios conmutadores. También puede ser más difícil rastrear el origen o destino de un mensaje si pasa por varias manos.

- ¿Qué posibilidades incluye la introducción de un conmutador en el sistema?
  En primer lugar se tiene que el conmutador puede organizar el flujo de información entre clientes. Además, hay una flexibilidad en la entrega de mensajes, pues los clientes no tienen que estar conectados al mismo tiempo. Finalmente se puede añadir filtros para ver qué se reenvía y qué no.

Explicar/Detallar la forma/protocolo que utilizaron para comunicarse en la parte del conmutador. Es decir, cómo determinaron el destino del mensaje, cómo determinaron una forma de no sobrecargar a su conmutador, etc.

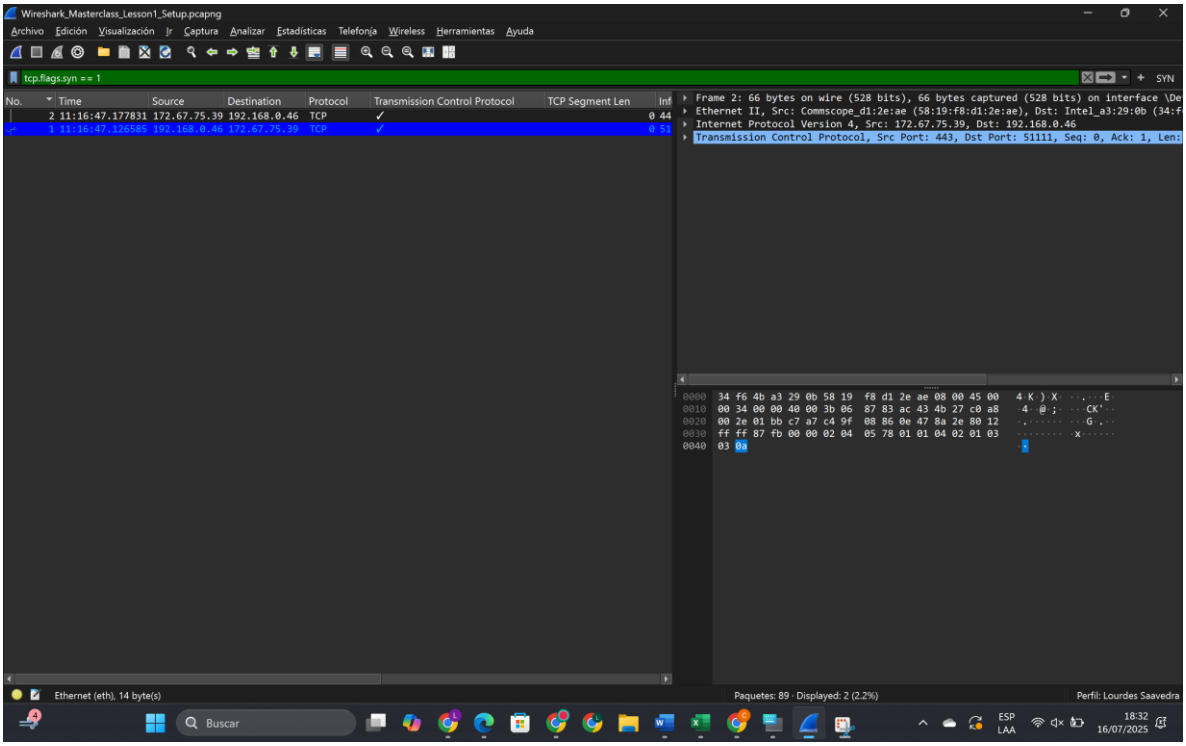
Durante esta parte diseñamos un protocolo de comunicación sencillo que permitiera organizar el envío de mensajes entre clientes sin generar confusión ni sobrecargar

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.

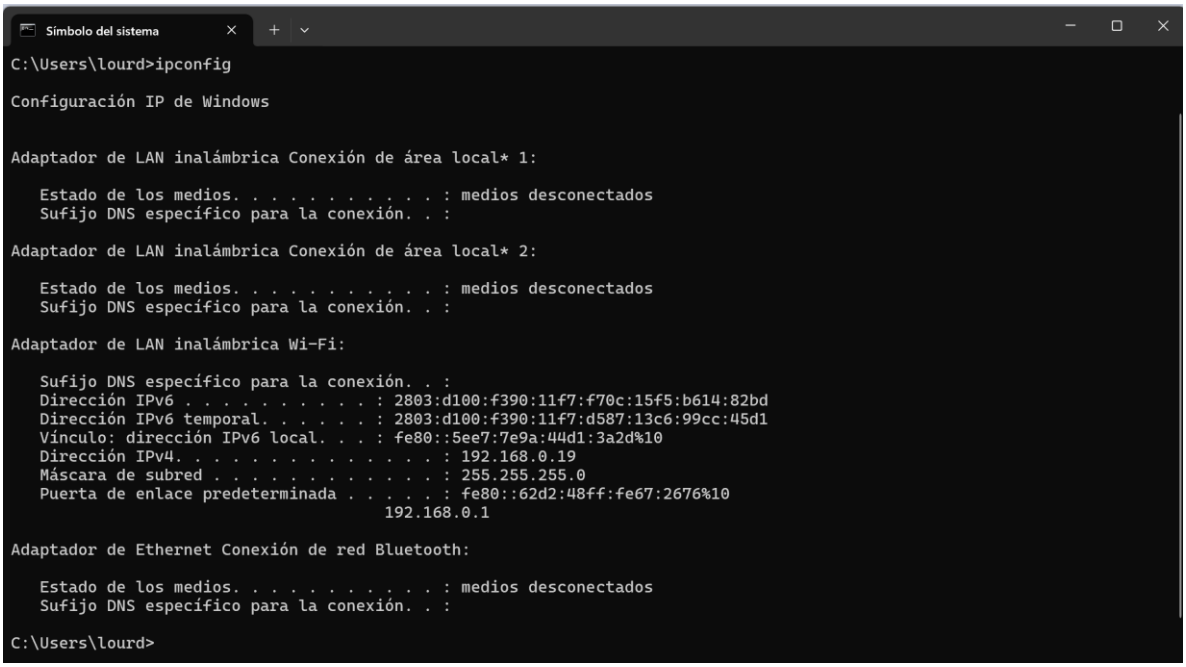
Además, se contemplaron medidas para el manejo de errores y congestión. En caso de múltiples envíos simultáneos, los mensajes se procesaban en orden de llegada y se notificaba a los clientes que esperaran. Si el mensaje era confuso o incompleto, el conmutador pedía una repetición, y una vez reenviado, 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.

Capturas y evidencias de la segunda parte

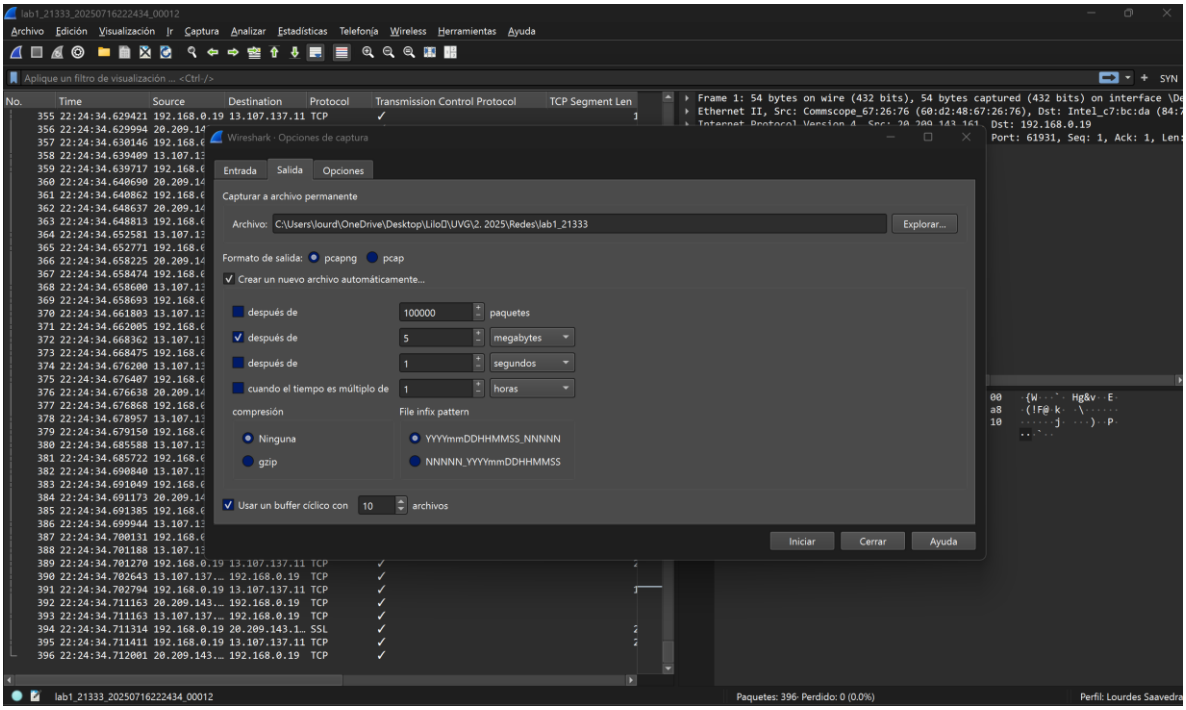
Personalización del entorno



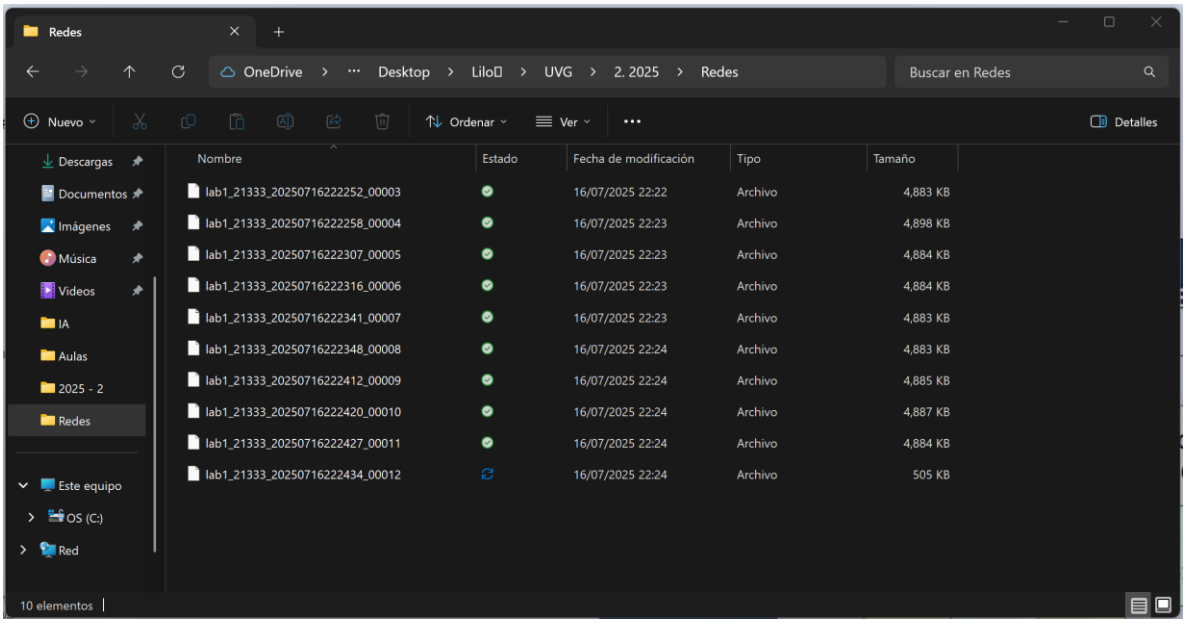
Configuración de la captura de paquetes



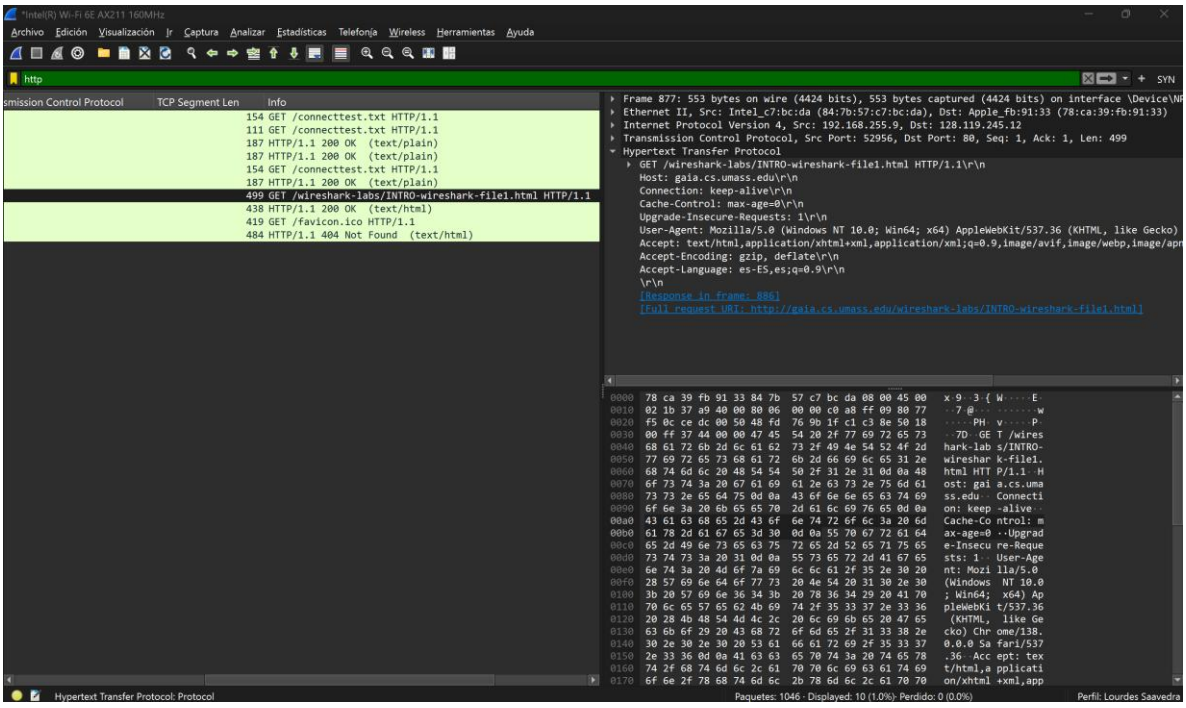
- ¿Cuál es su interfaz de red?
  Al ejecutar ipconfig, se observan varias interfaces de red, pero solo el adaptador de LAN inalámbrica Wi-Fi está activo, con la dirección IPv4 192.168.0.19. Las demás interfaces están desconectadas. Por lo tanto, mi interfaz de red principal es la Wi-Fi.

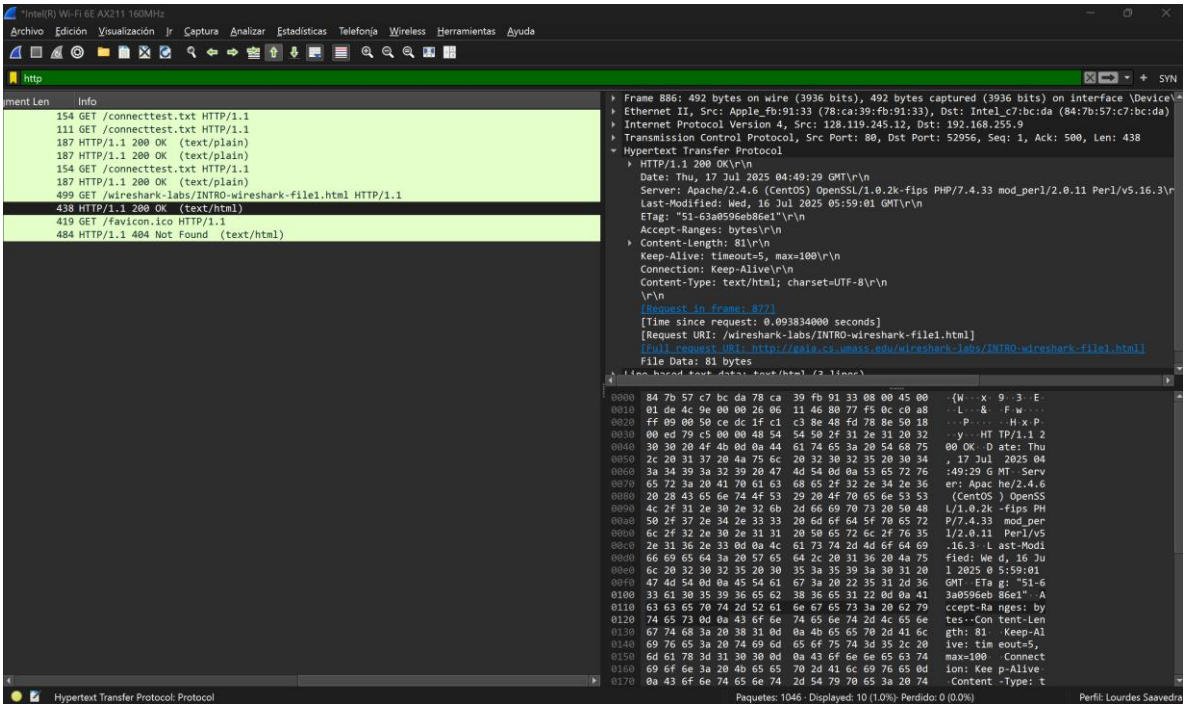


Lourdes Saavedra
21333



Análisis de paquetes





Respuestas a las preguntas de la segunda parte (a-e)

- ¿Qué versión de HTTP está ejecutando su navegador?
  Esto se puede ver con la línea: GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1. La versión de HTTP utilizada por el navegador es HTTP/1.1.

- ¿Qué versión de HTTP está ejecutando el servidor?
  Esto se puede ver también con GET. Entonces, el servidor también está usando HTTP/1.1.

- ¿Qué lenguajes (si aplica) indica el navegador que acepta a el servidor?
  Esto se puede ver con: Accept-Language: es-ES,es;q=0.9. Por lo que el navegador acepta contenido en español de España (es-ES) con preferencia alta y español genérico (es) como alternativa.

- ¿Cuántos bytes de contenido fueron devueltos por el servidor?
  Esto se puede ver con la línea: Content-Length: 81. Por lo que el servidor devolvió 81 bytes de contenido.

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

En caso de problemas de rendimiento, sería útil capturar paquetes tanto en el cliente como en el punto de acceso o router, para evaluar si el problema ocurre al enviar la solicitud o al recibir la respuesta. Ahora, no siempre es conveniente instalar Wireshark en el servidor, especialmente si es remoto o de terceros. En entornos controlados como servidores propios sí puede ser útil para detectar cuellos de botella, retrasos o retransmisiones.

Discusión sobre la actividad, su experiencia y hallazgos. Incluir ambas partes

Esta práctica permitió comprender la evolución de los sistemas de comunicación y la importancia de los protocolos para transmitir información de manera precisa. En la primera parte, al simular una red humana usando códigos como Morse y Baudot, se pudo observar la dificultad que implica transmitir datos sin medios digitales, especialmente cuando se depende únicamente del oído humano. El código Morse resultó ser más intuitivo, gracias a sus señales auditivas diferenciadas y a la claridad de pausas entre letras. En cambio, el Baudot, al requerir secuencias exactas de bits, generó más errores debido a confusiones o mala sincronización.

La fase de comunicación "empaquetada", realizada mediante notas de voz, introdujo desafíos adicionales relacionados con la asincronía y la falta de retroalimentación inmediata. Esto reforzó la necesidad de establecer reglas claras para estructurar mensajes y confirmó por qué los protocolos de red incluyen mecanismos de control y verificación.

En la parte del conmutador, se experimentó con la idea de intermediar mensajes a través de una persona designada. Esta dinámica permitió observar tanto la utilidad de los conmutadores para organizar el flujo de información, como los posibles cuellos de botella o retrasos que pueden surgir si no se establece un protocolo de control adecuado.

En la segunda parte, al trabajar con Wireshark, se observaron conocimientos técnicos sobre análisis de red. Personalizar la herramienta, aplicar filtros y estudiar el protocolo HTTP permitió visualizar cómo se comunican los dispositivos en una red real. Uno de los hallazgos más importantes fue comprobar que el navegador y el servidor usaban HTTP/1.1, que el cliente especifica preferencias de lenguaje y que es posible identificar otro tipo de propiedades. También fue valioso entender cuándo y dónde conviene capturar paquetes para diagnosticar problemas de rendimiento.

Comentarios

La práctica fue interesante, ya que combinó aspectos históricos, humanos y técnicos de las redes. Fue útil comenzar con una simulación tangible antes de pasar al análisis con herramientas especializadas. Considero que este enfoque facilitó el aprendizaje y generó conciencia sobre los múltiples factores que intervienen en una comunicación digital efectiva. Aunque hubo algunas dificultades técnicas con la captura de paquetes y el uso de Wireshark (en general), los pasos para solucionarlas reforzaron el entendimiento práctico del entorno de red y de Wireshark.

Conclusiones

Esta práctica permitió entender que la comunicación entre dispositivos requiere no solo de canales efectivos, sino también de estructuras lógicas y protocolos estandarizados. A través de la red humana, se evidenciaron las limitaciones del envío sincrónico de datos y se vio la importancia de elementos como confirmación, direccionamiento y gestión de errores. La introducción a Wireshark ofreció una ventana clara al tráfico real en una red, permitiendo interpretar cómo se da una solicitud HTTP desde el cliente hasta el servidor y de regreso.

Referencias

Wireshark Foundation. (s.f.). *Wireshark User's Guide*. https://www.wireshark.org/docs/