

Diederich Solis - 22952

Antecedentes

Previo a las redes de computadoras, como las conocemos hoy en día, la transmisión de información era de persona a persona. Sus primeros y más grandes usos fueron el telégrafo y el teléfono. Más adelante, con el objetivo de mejorar la tasa de transmisión, se aprovecharon sistemas automatizados que pudieran no solo controlar las rutas para llamadas telefónicas o la comunicación de datos digitales.

Por otra parte, es importante analizar y entender los paquetes que se transmiten durante la comunicación entre dispositivos. Los analizadores de red como Wireshark ayudan a realizar esta tarea y son una de las principales herramientas utilizadas por los administradores de redes, hackers éticos, etc.

Es por ello que realizaremos ciertas actividades introductorias a la ciencia de enviar información que nos permitirán percatarnos de muchos detalles, y también aprenderemos a configurar y utilizar Wireshark y sumarnos a la comunidad de científicos de computación, administradores y analistas de red, hackers, etc., que la utilizan, con el fin de asegurar el óptimo rendimiento y seguridad de nuestras redes.

Objetivos

- Identificar ventajas y desventajas de distintos esquemas de comunicación.
- Comprender e identificar la complejidad al momento de enviar información.
- Conocer las bases de un conmutador a pequeña escala.
- Conocer los propósitos y usos de una analizador de paquetes
- Familiarizarse con el entorno de Wireshark
- Fortalecer la teoría sobre paquetes a través del análisis de paquetes reales

Desarrollo

El laboratorio cuenta con dos partes principales. La primera parte se debe hacer de forma presencial/sincrónica para contar con sus parejas. La segunda parte puede realizarse en clase, o bien de manera asincrónica/de tarea. Al final, el documento/reporte a entregar lo trabajan y entregan de forma individual.

Red Humana (parte grupal en clase)

Existen distintas formas en que se puede representar la información al momento de enviarla a través de un medio. Un esquema utilizado en los tiempos de la telegrafía fue el código Morse, el cual podía representar, a través de pulsos, todas las letras del alfabeto en inglés y los 10 dígitos arábigos (ver **Imagen 1**). No obstante, con mejoras en los sistemas de telégrafo y su parcial automatización se introdujeron nuevos sistemas que podían aprovechar los avances tecnológicos. Uno de ellos fue el código de Baudot, el cual podía codificar todas las letras del alfabeto inglés, junto con códigos de control, a través de representaciones binarias de 5 bits (ver **Imagen 2**).

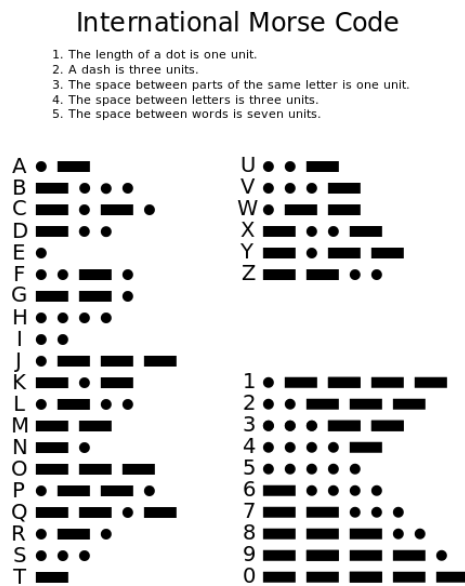


Imagen 1: Código Morse internacional.

Fuente

https://en.wikipedia.org/wiki/File:International_Morse_Code.svg

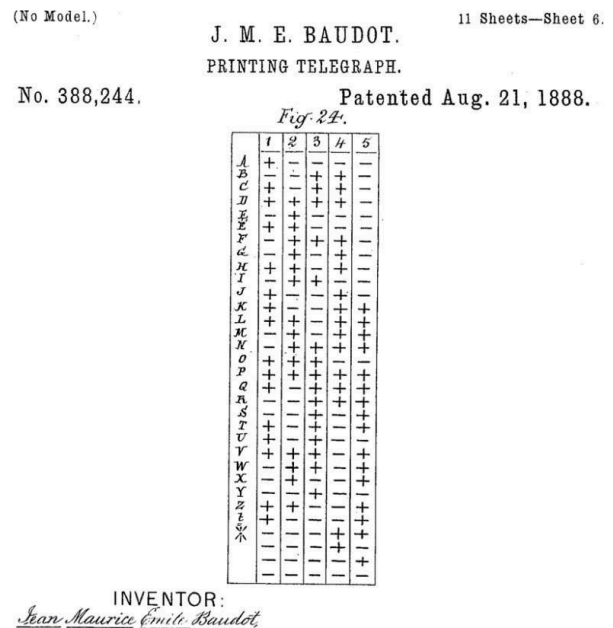


Imagen 2: Código de Baudot.

Fuente

https://en.wikipedia.org/wiki/Baudot_code#/media/File:Baudot_Code_-_from_1888_patent.png

Para el laboratorio se estarán enviando estos mensajes a través de un medio. Debido al factor remoto/híbrido estaremos haciendo una versión un poco más “moderna” de la actividad: enviaremos los mensajes emulando con nuestra voz, o con un objeto (i.e.: un lapicero en su escritorio), un generador de audio, etc., los elementos de nuestros mensajes. En el caso del código Morse pulsos (con duración variable) y en el caso del código de Baudot bits (1's y 0's).

La forma de envío de los mensajes depende de cada una de las partes de la actividad, como se detalla a continuación...

1.1 Primera parte: transmisión de códigos

En la primera parte se estarán distribuyendo en **parejas** (o un trío en caso de grupo impar). Cada pareja deberá practicar el envío y la recepción de mensajes utilizando los dos esquemas. Intentar enviar **al menos tres mensajes distintos (de 10 caracteres mínimo) por persona**, por cada uno de los esquemas. La comunicación se hará en un **Room de Zoom con su pareja, usando su micrófono** para enviar los mensajes.. **En caso presencial, la comunicación se hará directamente** (frente a frente, hablado). Durante la actividad, tengan en mente las siguientes preguntas (las debe incluir en su reporte):

Mensajes morse:

H →

O → ---

L → .-..

A → .-

Resultado: --- .-.. .-

A → .-

Y → -.-

U → ..-

D → -..

A → .-

Resultado: .- -.- ..- -.. .-

T → -

E → .

(space) → /

A → .-

M → --

O → ---

Resultado: - . / .- ---

Mensajes Baudot:

LTRS (modo letras): 11111

H → 10100

O → 01100

L → 10010

A → 00011

Resultado: 11111 10100 01100 10010 00011

LTRS: 11111

A → 00011

Y → 10111

U → 00110

D → 00100

A → 00011

Resultado: 11111 00011 10111 00110 00100 00011

LTRS: 11111

T → 10000

E → 00101

(space no se codifica directamente en Baudot, puede omitirse o usarse como separador externo)

A → 00011

M → 10011

O → 01100

Resultado: 11111 10000 00101 [espacio] 00011 10011 01100

- ¿Qué esquema es más fácil? ¿Más difícil?

Fácil: Código Morse

Patrones auditivos intuitivos: Los puntos (•) y rayas (–) son fáciles de vocalizar (ej: "dit" para •, "dah" para –) y distinguir auditivamente.

Diseño para comunicación humana: Está optimizado para transmisión oral, con ritmos reconocibles (ej: la letra E es un solo "dit", la T es una "dah").

Flexibilidad en el ritmo: Permite ajustar la velocidad de envío sin perder claridad (ej: pausas más largas entre letras si es necesario).

Difícil: Código Baudot

Estructura binaria rígida: Cada carácter requiere transmitir exactamente 5 bits (ej: A = 11000). Un error en un bit cambia completamente el carácter.

Sincronización crítica: El receptor debe identificar con precisión el inicio/fin de cada bit y carácter. Si se pierde la sincronización (ej: por una pausa mal calculada), se altera toda la secuencia.

- ¿Con cuál ocurren menos errores?
En lo personal es el morse, ya que los participantes lograrán transmitir mensajes con pocos errores después de 1-2 intentos. Errores comunes incluirán confundir letras con patrones similares en cambio con Los errores serán frecuentes (especialmente en bits individuales o shifts), y la decodificación será lenta.

1.2 Segunda parte: transmisión “empaquetada”

En la segunda parte repetiremos la dinámica anterior utilizando únicamente el esquema que más se les haya facilitado. En este caso, el envío se hará de una forma diferente: **mediante notas de voz (VN) enviadas por Whatsapp/Discord/etc. donde se graben ustedes emitiendo el mensaje en código. Deben intentar enviar al menos tres mensajes (de 10 caracteres) por persona, diferentes a los mensajes anteriores.** Durante la actividad, tengan en mente lo siguiente:

Mensajes morse:

H →

O → ---

L → .-..

A → .-

Resultado: --- .-.. .-

A → .-

Y → -.--

U → ..-

D → -..

A → .-

Resultado: .- -.-- ..- -.. .-

T → -

E → .

(space) → /

A → .-

M → --

O → ---

Resultado: - . / .- --- ---

Mensajes Baudot:

LTRS (modo letras): 11111

H → 10100

O → 01100

L → 10010

A → 00011

Resultado: 11111 10100 01100 10010 00011

LTRS: 11111

A → 00011

Y → 10111

U → 00110

D → 00100

A → 00011

Resultado: 11111 00011 10111 00110 00100 00011

LTRS: 11111

T → 10000

E → 00101

(space no se codifica directamente en Baudot, puede omitirse o usarse como separador externo)

A → 00011

M → 10011

O → 01100

Resultado: 11111 10000 00101 [espacio] 00011 10011 01100

- ¿Qué dificultades involucra el enviar un mensaje de esta forma “empaquetada”?
 - En Código Morse, emitir puntos (.) y rayas (-) por voz puede ser confuso si no se hace con buen ritmo o claridad.
 - En Baudot, decir cadenas de bits (como 1 0 1 1 0) puede ser tedioso y propenso a errores.
 - La entonación, pausas y duración de los sonidos son fundamentales para no perder el sentido.
 - El receptor debe entender el código y estar concentrado para interpretarlo correctamente.
 - Puede haber malentendidos si los sonidos no son claros o si se omite una pausa importante.
 - El receptor también debe saber en qué modo está el mensaje (letras o números, como en Baudot).
 - Las **notas de voz** pueden tener:
 - Ruido de fondo
 - Baja calidad del audio
 - Cortes o distorsiones
 - Si el mensaje se transmite por una red con **latencia o compresión**, esto puede alterar los sonidos.

1.3 Tercera parte: conmutación de mensajes

En la tercera parte la clase se repartirá cooperando con otra pareja/grupo, con quienes deberán determinar lo siguiente:

- Tres+ personas serán los clientes del servicio
- Una persona funcionará como conmutador

Entre ustedes se organizarán según la Topología que se muestra en la **Imagen 3**.

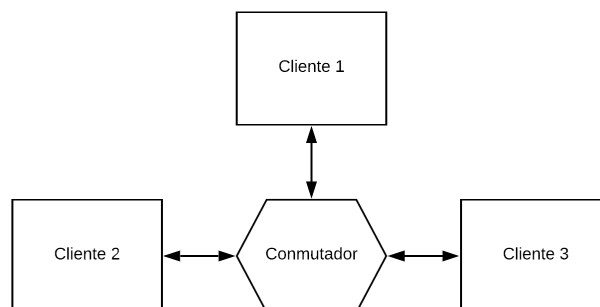


Imagen 3: Comunicación entre clientes y conmutador.

El conmutador recibirá la VN de Whatsapp/Discord de cualquiera de los clientes y luego lo estará reenviando al destino final. Para ello, deben de acordar cómo dirán al conmutador quién es el destino final del mensaje, así como determinar si el conmutador está listo o no para recibir mensajes. Durante la actividad, tengan en mente lo siguiente.

- ¿Qué posibilidades incluye la introducción de un conmutador en el sistema?

La incorporación de un conmutador en un sistema de comunicación introduce varias posibilidades que mejoran su rendimiento y organización. Un conmutador actúa como un nodo central de distribución, permitiendo gestionar y direccionar los mensajes entre los diferentes usuarios o dispositivos sin que tengan que comunicarse directamente entre sí.

Entre las posibilidades más importantes se incluyen:

- Distribución de carga: El uso de uno o varios conmutadores permite repartir el tráfico de mensajes, evitando sobrecargar a los nodos extremos y haciendo el sistema más eficiente.
- Escalabilidad: Facilita la incorporación de más usuarios o nodos al sistema sin afectar significativamente el rendimiento general.
- Organización del flujo de información: Se puede establecer un orden claro en la entrega de mensajes, evitando colisiones o interferencias.
- Mayor complejidad: Requiere una mayor coordinación y control para evitar errores en la entrega de mensajes, especialmente si hay múltiples conmutadores o rutas posibles.
- Necesidad de gestión de turnos y rutas: Si no se administra correctamente, puede haber confusión en los destinatarios, duplicación de mensajes o pérdida de información.

- ¿Qué ventajas/desventajas se tienen al momento de agregar más conmutadores al sistema?

Ventajas:

- Centralización y control: Un conmutador actúa como punto de control que dirige los mensajes al destinatario correcto, reduciendo la complejidad en los extremos.
- Simplicidad para los usuarios: Los usuarios no necesitan conocer la dirección de cada otro nodo, solo deben comunicarse con el conmutador.
- Optimización del tráfico: Puede priorizar mensajes, gestionar colas y mantener el orden en la red.
- Facilita la expansión: Es más fácil integrar nuevos nodos sin necesidad de crear múltiples conexiones directas.

Desventajas:

- Punto único de fallo: Si el conmutador central falla, toda la red podría verse afectada o quedar inoperativa.

- **Riesgo de congestión:** Si no se dimensiona correctamente, el conmutador puede saturarse y provocar retrasos o pérdida de mensajes.
- **Mayor complejidad técnica:** Su implementación y mantenimiento requieren mayor conocimiento técnico y recursos.
- **Vulnerabilidad en la seguridad:** Al centralizar la comunicación, se convierte en un objetivo sensible ante posibles ataques o errores.

Introducción a Wireshark (parte individual de tarea)

Se debe descargar e instalar el software de [Wireshark](https://www.wireshark.org/). Es probable que para ejecutarlo pida permisos de administrador (sudo, click + run as admin, etc.).

1.1 Primera parte: personalización del entorno

En la primera parte se realizará la personalización del entorno de Wireshark, de modo que se adapte a nuestras preferencias de uso.

1. Descargue el archivo <https://www.cloudshark.org/captures/e6fb36096dbb> (Export -> Download)
2. Cree un perfil (Configuration -> Profiles) con su primer nombre y primer apellido
3. Abra el archivo descargado (File -> Open)
4. Aplique el formato de tiempo Time of Day a la columna Tiempo (View -> Time Display)
5. Agregue una columna con la longitud del segmento TCP (Selecciona la primera fila, en el panel inferior despliegue Transmission Control Protocol, seleccione TCP Segment Len y aplíquelo como una columna)
6. Elimine u oculte la columna Longitud
7. Aplique un esquema de paneles que sea de su preferencia (que no sea el esquema por defecto)
8. Aplique una regla de color para los paquetes TCP cuyas banderas SYN sean iguales a 1, y coloque el color de su preferencia (View -> Coloring Rules)
9. Cree un botón que aplique un filtro para paquetes TCP con la bandera SYN igual a 1.
10. Oculte las interfaces virtuales (en caso aplique)

Se debe realizar tomas de pantalla que muestren el entorno final personalizado, el nombre del perfil y el uso de la regla de color y botón del filtro, así como la lista simplificada de las interfaces de captura.

1.2 Segunda parte: configuración de la captura de paquetes

En la segunda parte, se realizará una captura de paquetes con un ring buffer.

1. Abra una terminal y ejecute el comando `ifconfig/ipconfig` (dependiendo de su SO). Detalle y explique lo observado, investigue (i.e.: 'man ifconfig', documentación) de ser necesario. ¿Cuál es su interfaz de red?
2. Luego, retornando a Wireshark, desactive las interfaces virtuales o que no aplique.
3. Realice una captura de paquetes con la interfaz de Ethernet o WiFi con una configuración de ring buffer, con un tamaño de 5 MB por archivo y un número máximo de 10 archivos (Capture -> Options -> Output) Genere tráfico para que los archivos se creen. Defina el nombre de los archivos de la siguiente forma: `lab1_carnet.pgcap`

Se debe realizar tomas de pantalla de la configuración o comandos para la creación del ring buffer, así como los archivos generados.

1.3 Tercera parte: análisis de paquetes

En la tercera parte se analizará el protocolo HTTP. Debe realizar tomas de pantalla que validen sus respuestas.

1. Abra su navegador, inicie una captura de paquetes en Wireshark (sin filtro) y acceda a la siguiente dirección (Si por alguna razón debe repetir el paso, borre su caché o utilice el modo incógnito de su navegador):
<http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html>
 2. Detenga la captura de paquetes (si desea realizar una nueva captura de la página deberá borrar el caché de su navegador, de lo contrario no se realizará la captura del protocolo HTTP).
 3. Responda las siguientes preguntas:
 - a. ¿Qué versión de HTTP está ejecutando su navegador?
 - b. ¿Qué versión de HTTP está ejecutando el servidor?
 - c. ¿Qué lenguajes (si aplica) indica el navegador que acepta al servidor?
 - d. ¿Cuántos bytes de contenido fueron devueltos por el servidor?
 - e. 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.
-

Reporte

Al finalizar la actividad debe realizarse un **reporte individual** donde se incluyan las siguientes secciones:

- Nombre y carnet / Nombre y carne de pareja
- Nombres y carnet de la otra pareja
- Título de la práctica
- Descripción de la práctica
- Respuestas de las preguntas de la primera parte:
 - o ¿Qué esquema (código) fue más fácil de transmitir y por qué? ¿Qué esquema (código) fue más difícil de transmitir y por qué?
 - o ¿Qué esquema tuvo menos errores (incluir datos que lo evidencien)?
 - o ¿Qué dificultades involucra el enviar un mensaje de forma “empaquetada”?
 - o ¿Qué ventajas/desventajas se tienen al momento de agregar más conmutadores al sistema?
 - o ¿Qué posibilidades incluye la introducción de un conmutador en el sistema?
- 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.
- Capturas y evidencias de la segunda parte.
- Respuestas a las preguntas de la segunda parte (a-e).
- Discusión sobre la actividad, su experiencia y hallazgos. Incluir ambas partes.
- Comentarios
- Conclusiones
- Referencias Utilizadas

Rúbrica de evaluación

Elemento	Ponderación
Parte 1 (En Clase, borrador)	50%
	<ul style="list-style-type: none">• Descripción de la Práctica e Intro/Antecedentes• Respuestas a las preguntas en el documento.• Explicar su protocolo en la parte 1.3 (conmutador)• etc.
Parte 2 (para llevar, reporte final)	50%
	<p>La segunda entrega contiene la versión final del documento, que incluye y se trabaja sobre el borrador de la primera entrega. Algunas cosas a incluir en la segunda parte incluyen:</p> <ul style="list-style-type: none">• Formato adecuado con encabezado, descripción, etc.• Capturas de paquetes y evidencia de la segunda parte• Respuestas a las preguntas en el documento de la segunda parte (a-e)• Discusión de la actividad y sus comentarios• Conclusiones, Referencias, etc.

Entregar en Canvas

- Link a su repositorio donde tendran varios commits, por lo menos uno por entrega.
Incluyen en su repo **y** en Canvas:
 - Archivo .pdf con su reporte **individual**, debidamente identificado y con buen formato de reporte.
 - Todo archivo generado, como el .pgcap o similares de Wireshark.
 - Cualquier otro archivo necesario para calificar su trabajo.